POLITECHNIKA ŚLĄSKA BIBLIOTEKA GŁÓWNA w Gliwicach
P. 1877/75

9 1975

informatyka

SPIS TREŚCI NR 9/75

Systemy komputerowe ODRA 1300 i JS 1032 roku 1975
Thanasis Kamburelis, Andrzej Zasada 1

Trendy w stosowaniu języków programowania
Zdzisław Dybizbański 6

Elementy strategii automatyzacji prac projektowych
Włodzimierz Trafalski 9

Normatywy programowania
Aleksander Kuźmin 14

SESTEL system ewidencji środków telekomunikacji
Krzysztof Gajos 16

KADRY INFORMATYKI

Kształcenie kadr na poziomie średnim
Bogumił Stachura 18

Szkolenie projektantów systemów w firmie ICL
Jerzy Choroszko 21

TRYBUNA CZYTELNIKA

Kilka uwag na temat słownictwa informatycznego — *Tomasz Rawiński* 23

Z KRAJU

Dokąd zmierzasz polska teleinformatyko? 25

Refleksje nad stanem programowania — *Krystyn Bernatowicz* 27

ZE ŚWIATA

HONYWELL-BULL dokonuje fuzji z CII — *Jerzy Miłek* 27

Wersja dyskowa minikomputera SINGER 1500 — *Władysław S. Gajewski* 29

Minikomputer ogólnego przeznaczenia jako procesor czołowy — *Jerzy Kurkiewicz, Stanisław Michalski* 32

Informacje różne 33 i 34

TERMINOLOGIA ZE ZJEDNOCZENIA INFORMATYKI

Terminologia — oprac. *Władysław Klepacz* 35

Białowieża, Kołobrzeg i co dalej? Refleksje uczestnika *Jerzy Dańda, Irena Malerczyk-Dańda* 36

Seminarium: Organizacja, metodyka i technika produkcji oprogramowania. Kołobrzeg 14—17.IV. 1975 — *Jerzy Kurkiewicz* 37

OŚRODKI INFORMATYKI PREZENTUJĄ

I moduł systemu „Gospodarka materiałowa", „Ewidencja obrotów i stanów magazynowych" — *Witold Jędrzejewski* 39

SZTUKA KOMPUTEROWA

Komputery w muzyce (4) — *Marek Hołyński* 40

PROBLEMATYKA BAZY DANYCH

Ogólne problemy organizowania bazy danych w systemie DMS-2 — *Andrzej Kuroń* 43

PRZEGLĄD WYDAWNICTW

Bibliografia wydawnictw polskich z dziedziny informatyki — oprac. *Jerzy Klamborowski* 47

Ogłoszenia III i IV okł.

WYDAWNICTWA CZASOPISM TECHNICZNYCH NOT
Warszawa
Czackiego 3/5

KOLEGIUM REDAKCYJNE

Redaktor Naczelny prof. dr hab. Leon ŁUKASZEWICZ

mgr Krystyn BERNATOWICZ, prof. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), doc. dr inż. Zbigniew GACKOWSKI, mgr inż. Marek HOŁYŃSKI, Władysław KLEPACZ, (zast. redaktora naczelnego) doc. dr hab. Antoni MAZURKKIEWICZ

Sekretarz Redakcji mgr Krystyna WROŃSKA Red. tech. Józef DUSZA

RADA PROGRAMOWA

Mgr inż. Antoni BOSSOWSKI, doc. dr inż. Jan FELICKI, doc. dr inż. Zbigniew GACKOWSKI, doc. dr inż. Aleksander GOLINOWSKI, dr hab. Andrzej GRZYWAK, doc dr hab. inż. Roman KULESZA, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. inż. Stanisław PASZKOWSKI (wiceprzewodniczący), prof. dr Tadeusz PECHE, mgr inż. Bronisław PIWOWAR, dr inż. Andrzej PLĄSKOWSKI, mgr inż. Tadeusz PODGÓRSKI (wiceprzewodniczący), prof. dr inż. Jerzy SEIDLER, prof. dr inż. Andrzej STRASZAK (przewodniczący), doc. Jerzy TRYBULSKI, dr Tadeusz WALCZAK, prof. dr inż. Stefan WĘGRZYN, dr inż. Jan Z. ŻYDOWO

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pokój 331, tel. 27-71-40 lub centrala 28-82-61 w. 285, dyżury redakcji 10,00—13,00

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka", Z. 2. Zam. 334. Papier druk. sat IV kl. 65 g 61 × 86. Obj: 6 ark. druk. Nakład 6100. B-89.

Cena egzemplarza zł 15.— INDEKS 36210/36124 Prenumerata roczna zł 180.—

Informatyka

dawniej Maszyny Matematyczne

zastosowania w gospodarce, technice i nauce

Nr 9

MIESIĘCZNIK

1975

ROK XI

Wrzesień

P. 1877/75

ORGAN MINISTERSTWA NAUKI, SZKOLNICTWA WYŻSZEGO I TECHNIKI I POLSKIEGO KOMITETU AUTOMATYCZNEGO PRZETWARZANIA INFORMACJI NACZELNEJ ORGANIZACJI TECHNICZNEJ

THANASIS KAMBURELIS
ANDRZEJ ZASADA
MERA-ELWRO
Wrocław

Systemy komputerowe ODRA 1300 i JS 1032 roku 1975

W artykule przedstawiono aktualnie produkowane przez zakłady MERA ELWRO we Wrocławiu systemy komputerowe ODRA 1300 (ODRA 1304, 1325 i 1305) oraz JS 1032. Omówiono najważniejsze zagadnienia związane z architekturą logiczną, konstrukcją, technologią i oprogramowaniem tych systemów.

Trochę historii na 15-lecie Zakładów MERA-ELWRO

Pierwsze komputery prodkucji Zakładów MERA-ELWRO pojawiły się w roku 1963. Były to maszyny UMC-1, tzw. pierwszej generacji, których konstrukcja oparta została o lampy elektronowe.

Od tamtych czasów wiele się zmieniło, zarówno w architekturze logicznej, oprogramowaniu, konstrukcji, technologii, jak i w metodach produkcji i eksploatacji komputerów.

W minionym okresie Zakłady MERA-ELWRO przeszły od komputerów pierwszej generacji — poprzez komputery drugiej generacji (np. ODRA 1204, 1304) — do komputerów trzeciej generacji ODRA 1305 i 1325 (opartych o układy scalone standardowej skali integracji) i rozpoczęły w bieżącym roku seryjną produkcję zmodernizowanej maszyny Jednolitego Systemu oznaczonej szyfrem JS-1032 (lub w skrócie R-32), w której zastosowano układy scalone o średniej skali integracji.

Struktura komputerów zmieniła się od szeregowych struktur logicznych do struktur równoległych, od maszyn jednoprogramowych do wieloprogramowych, od systemów jednomaszynowych do wielomaszynowych i wielodostępnych.

W przygotowaniu są również systemy z wielopoziomową organizacją pamięci i z dynamicznym podziałem pamięci, dzięki czemu systemy te staną się bardziej efektywne w działaniu i eksploatacji.

Szybkość komputerów produkcji MERA-ELWRO wzrosła w minionym okresie do ponad 6000 razy, zaś koszt wykonania miliona dodawań zmniejszył się z 500 zł do 0,0028 zł, czyli około 170 000 razy. Zmieniała się również struktura wartości średniej konfiguracji komputerowej. Pierwsze maszyny cyfrowe składały się głównie z jednostki centralnej, która stanowiła 94% wartości zestawu i z prostych urządzeń zewnętrznych (przeważnie dalekopis i urządzenia taśmy papierowej). Natomiast zestawy urządzeń zewnętrznych zwiększyły się w takim stopniu, że stanowią obecnie często ponad 70% wartości całej konfiguracji komputerowej, pomimo ogromnego wzrostu sprzętu samej jednostki centralnej.

Systemy komputerowe ODRA 1300

System komputerowy ODRA 1300, który składa się z jednostek centralnych ODRA 1304, 1325 i 1305 oraz z szeregu urządzeń zewnętrznych, jest dzisiaj podstawowym sprzętem, w oparciu o który realizowany jest krajowy program informatyki. Do końca br. Zakłady MERA-ELWRO wyprodukują około 330 zestawów komputerowych systemu ODRA 1300. Popularność tego systemu wciąż rośnie wśród użytkowników, pomimo, że jego początkowe koncepcje architektury logicznej powstały ponad 10 lat temu, wynika ona nie tylko z dobrych rozwiązań technicznych, ale również (a może przede wszystkim) z faktu, że system ten ma bardzo obszerne i zweryfikowane oprogramowanie podstawowe oraz oprogramowanie użytkowe do najrozmaitszych dziedzin zarządzania i planowania, automatyzacji prac projektowych, obliczeń naukowo-technicznych, sterowania procesami technologicznymi i innych.

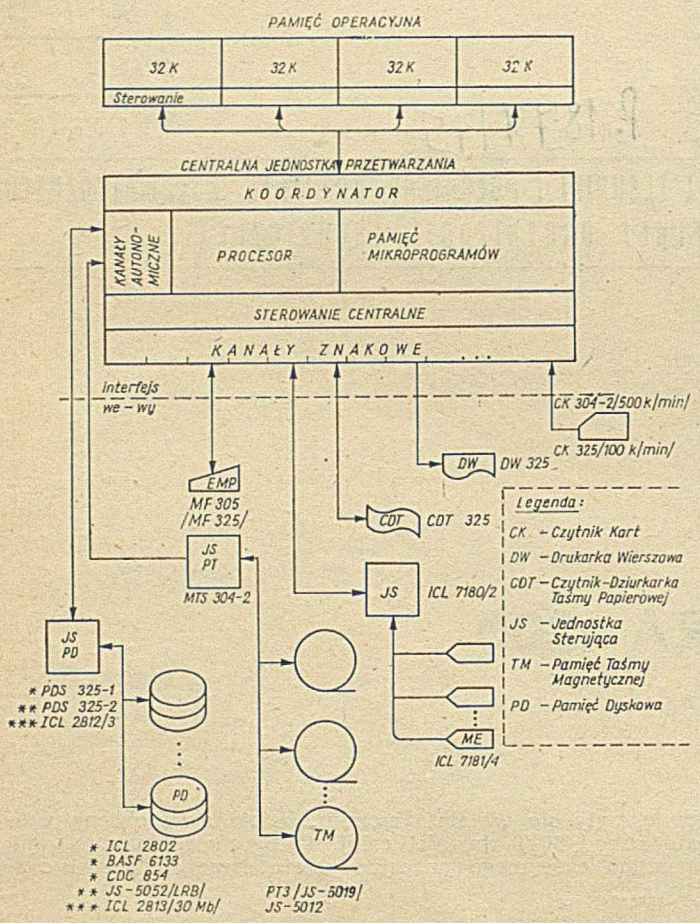
Sprzęt i środki programowe systemu ODRA 1300, opracowane przez Zakłady MERA-ELWRO oraz przez inne przedsiębiorstwa, instytuty lub wyższe uczelnie, pozwalają tworzyć w sposób elastyczny i ekonomiczny różne konfiguracje użytkowe zarówno do przetwarzania tradycyjnego, tj. lokalnego (rys. 1), jak i do przetwarzania zdalnego (rys. 2).

Jednolity System EMC

Jednolity System Elektronicznych Maszyn Cyfrowych (JS EMC), składający się z szeregu jednostek centralnych o zróżnicowanej mocy obliczeniowej, lecz o jednakowej architekturze logicznej i o wspólnych zasadach działania, oraz z bogatego zestawu urządzeń zewnętrznych, jest wynikiem współpracy naukowo-technicznej i przemysłowej krajów socjalistycznych.

Pierwszą polską maszyną Jednolitego Systemu jest maszyna cyfrowa R-32. Maszyna R-32 — podobnie jak i maszyna R-20, R-30, R-40 i R-50, należy pod względem funkcjo-

nalnym do tzw. pierwszego pokolenia maszyn Jednolitego Systemu (RIAD-I). Wyżej wymienione maszyny mają w pełni jednolitą architekturę logiczną, chociaż bazują na różnych rozwiązaniach konstrukcyjnych i różnych technolo-



Rys. 1. Komputer ODRA 1305(1325) — konfiguracja konwencjonalna

giach wytwarzania. Dzięki temu można było przyjąć również jednolite oprogramowanie na poziomie instrukcji maszynowych. Natomiast maszyny R-10 i R-20A nie są w pełni programowo zgodne z pozostałymi maszynami. Maszyna R-10 z powodu stosowania różnych instrukcji nielegalnych (uprzywilejowanych). Przyjęcie zgodnej architektury logicznej większości jednostek centralnych i urządzeń zewnętrznyh oraz jednolitego oprogramowania, stworzyło praktyczne możliwości skoncentrowania kadry naukowo-technicznej i potencjału przemysłowego krajów socjalistycznych (Bułgarii, Czechosłowacji, Niemieckiej Republiki Demokratycznej, Polski, Węgier i Związku Radzieckiego, a od początku 1974 r. także Kuby i Rumunii) nad realizacją wspólnego i ambitnego programu Elektronicznej Techniki Obliczeniowej, tj. Programu RIAD, który powinien w sposób optymalny zaspokoić potrzeby współpracujących krajów w zakresie informatyki.

Polska maszyna Jednolitego Systemu R-32 pojawiła się nieco później od pozostałych modeli systemu RIAD I (zademonstrowana została po raz pierwszy na MTP 1974), ale dzięki temu można było jej konstrukcję oprzeć o najnowsze krajowe rozwiązania konstrukcyjno-technologiczne oraz o elementy scalone średniej skali integracji typu TTL. Przyjęta baza technologiczna i elementowa pozwoliła uzyskać wysokie współczynniki techniczno-ekonomiczne (np. stosunek szybkości do kosztów wytwarzania). Dzięki wysokiej wydajności obliczeniowej, maszynę R-32 można zaliczyć do klasy średnich lub dużych jednostek centralnych, zależnie od rozmiarów pojemności pamięci operacyjnej oraz liczby kanałów przesyłania informacji. Szybkość maszyny wynosi około 500 000 dodawań na sekundę, zaś średnia szybkość według mieszanki Gibson I — ponad 200 000 operacji na sekundę: pojemność pamięci operacyjnej może się wahać w granicach od 128 Kb do 1024 Kb. (rys. 3. 4 i 5).

W maszynie R-32 wbudowano, jako standardowe, następujące środki funkcjonalne:

— środki wykonywania instrukcji standardowych (tj. operacji arytmetyki stałoprzecinkowej, logicznych, przesuwań, sterujących, operacji wejścia—wyjścia i innych), instrukcji arytmetyki dziesiętnej i arytmetyki zmiennoprzecinkowej

— szesnaście rejestrów ogólnych (akumulatorów) i cztery zmiennoprzecinkowe

— jeden kanał bajtowego multipleksora

— trzy kanały selektorowe

— środki ochrony pamięci operacyjnej

— środki bezpośredniego sterowania (do przekazywania lub odbierania danych bezpośrednio do urządzenia zewnętrznego lub do drugiego procesora)

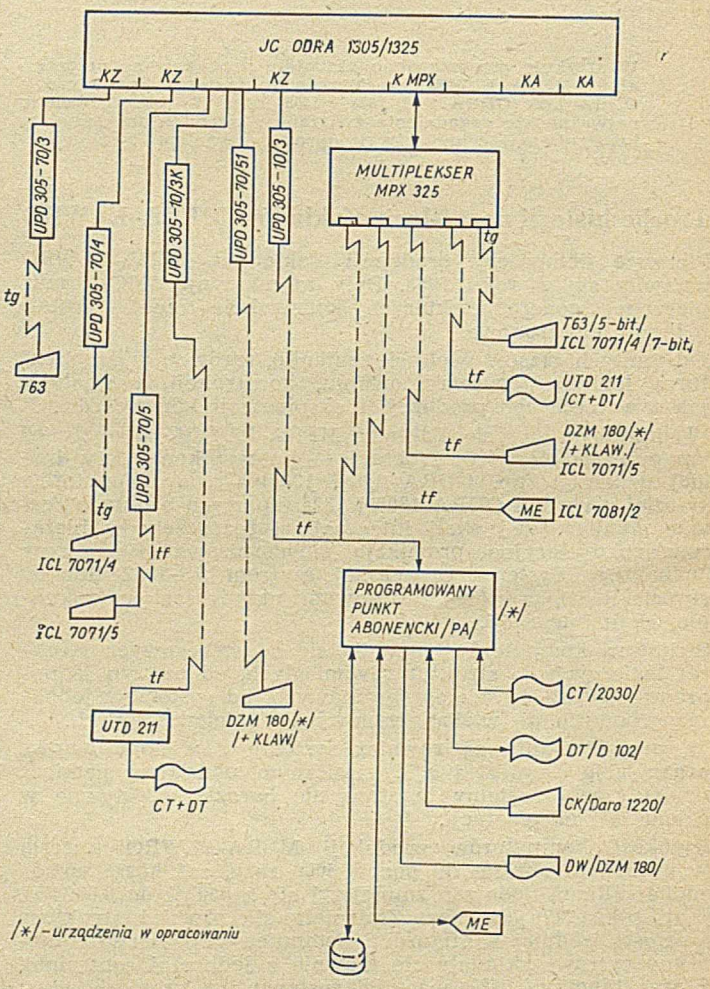
— czasomierz interwałowy i inne środki.

Przy opracowywaniu konstrukcji maszyny R-32 przyjęto jako generalną zasadę stosowania bazy elementowej oraz rozwiązań konstrukcyjnych i technologicznych wyprzedzających możliwości przemysłu krajowego o 2—3 lata, dzięki czemu w momencie rozpoczęcia produkcji seryjnej maszyna ta będzie technicznie nowoczesna. Dzisiaj można stwierdzić, że zasada ta okazała się bardzo trafna, gdyż wszystkie prawie zespoły przyjęte w konstrukcji maszynyn R-32, wchodzą dziś do seryjnej produkcji w PRL lub w innych krajach socjalistycznych.

W konstrukcji maszyny przyjęto następujące rozwiązania:

— elementy scalone średniej (i standardowej) skali integracji

— wielowarstwowe (do 12 warstw) płytki pakietów o dużym formacie



Rys. 2. Komputer ODRA 1305(1325) — konfiguracja teleprzetwarzania (w przygotowaniu)

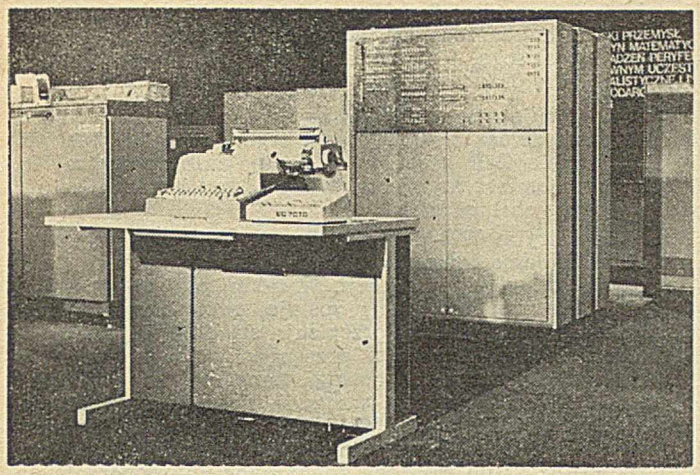


Foto 1. Jednostka centralna komputera R-32 wraz z konsolą EC-7070

— wielowarstwowe platery
— szybka pamięć mikroprogramów o dużej pojemności
— pojemna ferrytowa pamięć operacyjna o nowoczesnej konstrukcji (tzw. planarnej)
— płaskie kable do połączeń między blokami maszyny
— bezszafowa, aluminiowa konstrukcja mechaniczna.

W wyniku realizacji powyższych założeń maszyna R-32 charakteryzuje się następującymi cechami:
— wysoka łączna wydajność maszyny
— łatwość rozszerzania pojemności pamięci operacyjnej (do 1024 Kb)
— bardzo wysoki współczynnik wydajności — koszt (np. procesor i kanały zajmują tylko 29 pakietów logicznych o wymiarach 295 × 150 mm)
— wysoka niezawodność
— małe rozmiary jednostki centralnej (jak na maszynę klasy średniej)
— bardzo mały pobór mocy jednostki centralnej (z pamięcią o pojemności 256 Kb—3,6 KVA).

Pamięć lokalna maszyny (rys. 3), tj. pamięć robocza, zbudowana jest na układach scalonych i składa się z 64 słów 32-bitowych. Pamięć ta zawiera: 16 rejestrów ogólnych, 4 rejestry zmiennoprzecinkowe i 40 rejestrów robczych procesora i kanałów.

Pamięć mikroprogramów (PAM) jest pamięcią stałą typu transformatorowego i ma cykl równy 300 ns. Zawiera ona 2816 słów 86-bitowych, w których zakodowane jest zarówno sterowanie procesora jak i kanałów. Zastosowanie pamięci mikroprogramów o takiej pojemności pozwoliło na znaczne uprszczenie i zmniejszenie logiki maszyny, a także na wyposażenie maszyny w pewne nowe, w stosunku do innych maszyn JS EMC, własności jak:
— okresowa kontrola poprawności działania maszyny (włączana automatycznie)
— automatyczne zapamiętanie stanu maszyny w PAO przy wyłączeniu zasilania i odtwarzania tego stanu po włączeniu zasilania
— specjalne wbudowane testy techniczne procesora i pamięci.

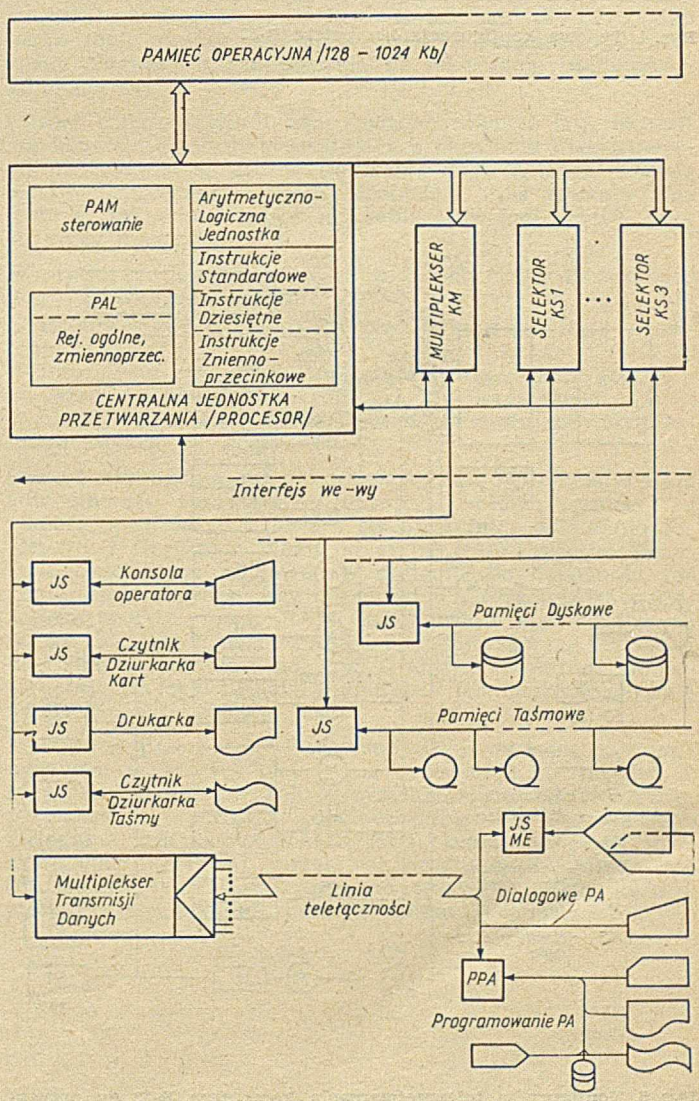
Pamięć mikroprogramów (stała) zbudowana jest w technice transformatorowej i rozmieszczona na płytkach 4-warstwowych o wielkości identycznej jak dla pakietów logicznych.

Całość pamięci stałej składa się z 16 pakietów o tej samej konstrukcji, różniących się między sobą tylko zawartością informacji „zaszytych" w rdzeniach ferrytowych. Każdy z pakietów zawiera 352 słowa 43-bitowe. Pamięć stała maszyn R-32 odznacza się znaczną szybkością działania (jak na technikę transformatorową): czas dostępu wynosi 120 ns, a czas cyklu 300 ns. Dzięki zastosowaniu oszczędnych układów sterowania — jeden układ wybierający steruje czterema słowami — uzyskano niewielkie rozmiary i niski koszt całej pamięci.

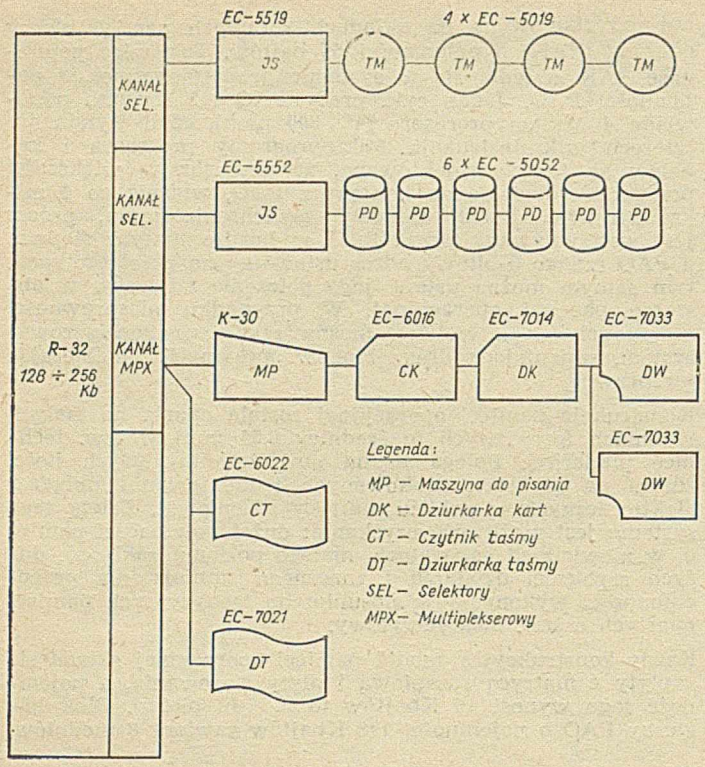
Pamięć operacyjna (PAO) składa się z bloków logicznych o pojemności 128 Kb. Każdy blok 128 Kb pamięci operacyjnej ma zbudowaną na układach scalonych pamięć kluczy ochrony (PAK) o pojemności 64 bajtów. Tak więc pojemność PAK zmienia się w systemie proporcjonalnie do pojemności PAO. Jeden cykl pracy PAO (1200 nsek). odpowiada 4 cyklom procesora (4 × 300 nsek), czyli wykonaniu czterech mikroinstrukcji. Mikroprogramy procesora i kanałów są tak zaprojektowane, że procesor w większości przypadków uruchamia kontakt z PAO, właśnie co 4 mikroinstrukcje. Dzięki takiemu zsynchronizowaniu, proces jest wstrzymywany przez PAO w bardzo małym stopniu, a PAO rzadko 3-bitowy adres ustawiany na przełącznikach, tym samym można ustalić jego położenie adresowe w obszarze pamięci operacyjnej. W przypadku niesprawności któregokolwiek z bloków, można szybko zrekonfigurować system, zmieniając adresy bloków sprawnych i odłączając niesprawne.

Konstrukcja pamięci operacyjnej została oparta na małych rdzeniach ferrytowych (o średnicy 0,54 mm) w tzw. technice planarnej. Polega to na umieszczaniu dużych ilości rdzeni na płytkach drukowanych i montowaniu układów elektronicznych w pobliżu matrycy rdzeniowej. Zaletą tego systemu jest możliwość uzyskania: dużej pojemności pamięci w niewielkich gabarytach, małego poziomu zakłóceń, dużych szybkości działania i znacznego zmniejszenia pracochłonności wykonania w stosunku do tradycyjnych pamięci opartych o tzw. blok ferrytowy.

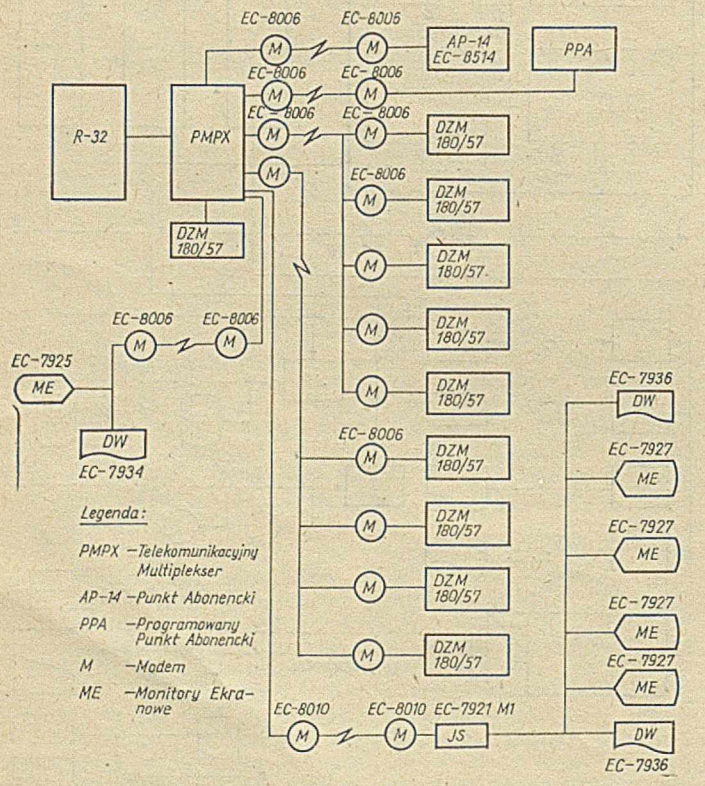
Każdy konstrukcyjny moduł pamięci operacyjnej składa się z płyty z matrycą rdzeniową i płyty sterowania, a pojemność jego wynosi 16 Kbajtów (8 K × 18 bitów). Blok logiczny PAO o pojemności 128 Kbajtów zawiera 8 modułów,



Rys. 3. Ogólna struktura komputera JS-1032 (R-32)



Rys. 4. Typowa konfiguracja komputera R-32



Rys. 5. Konfiguracja teleprzetwarzania komputera R-32 (w przygotowaniu)

1 pakiet sterowania i 3 pakiety adaptera. Moduły konstrukcyjne PAO zakończone są łączówkami i można je wymieniać między sobą, co znacznie przyspiesza wykrywanie uszkodzeń i naprawy.

Procesor stosuje mechanizmy i szyny 4-bajtowe. Pracuje on w jednym z dwóch trybów pracy: trybie CJP i trybie WE-WY. W trybie CJP procesor wykonuje mikroprogramy instrukcji, natomiast w trybie WE-WY realizowane są mikroprogramy obsługi kanałów.

Przejście z trybu WE-WY odbywa się w sposób wymuszony, tj. na żądanie kanałów, i może nastąpić w trakcie wykonywania dowolnej instrukcji. W takim przypadku mikroprogram bieżącej instrukcji zostaje zawieszony, a adres następnej mikroinstrukcji przepisuje się do specjalnego rejestru buforowego i procesor przechodzi do wykonania jednego z mikroprogramów obsługi kanałów. Po skończeniu operacji WE-WY, procesor wznawia zawieszony mikroprogram instrukcji. W trybie WE-WY określone części mikroinstrukcji dekodowane są odmiennie niż w trybie CJP. W ten sposób uzyskuje się dodatkowy zbiór mikrooperacji służących wyłącznie do sterowania pracą kanałów.

Sterowanie centralne rozwiązane zostało w taki sposób, aby cykl pracy procesora i kanałów ograniczony był tylko sumą propagacji sygnałów informacyjnych, natomiast samo sterowanie nie wnosi dodatkowych opóźnień. Uzyskano to dzięki zastosowaniu m.in. dodatkowego rejestru buforowego dla mikroinstrukcji pobieranych z pamięci stałej. W ten sposób procesor i kanały pracując z cyklem 300 ns wykorzystują maksymalną szybkość działania pamięci stałej.

Kanały wejścia—wyjścia współpracują z jednej strony z układami procesora i pamięcią mikroprogramów, z drugiej sterują pracą urządzeń zewnętrznych. W standardowej wersji maszyn R-32 montuje się jeden kanał multipleksorowy i trzy kanały selektorowe.

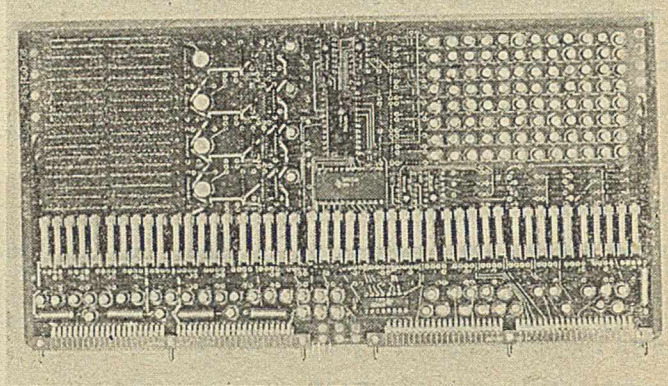


Foto 2. Pakiet modułu pamięci mikroprogramów (z transformatorami, matrycą diodową i tranzystorową)

Szybkość pracy kanału multipleksorowego wynosi 110 i 250 Kbajtów na sekundę, odpowiednio dla trybu multiplekserowego (w którym jednocześnie obsługuje się wiele urządzeń) i trybu selektorowego (w którym obsługuje się tylko jedno urządzenie).

Jako pamięć roboczą dla słów sterujących kanału multipleksorowego wykorzystuje się niedostępny programowo obszar PAO równy 2 lub 4 Kbajtom, co pozwala przechowywać słowa sterujące odpowiednio 128 lub 256 podkanałów. Obszar ten znajduje się zawsze na końcu zainstalowanej PAO. Takie rozwiązanie pamięci multipleksora jest bardzo proste i tanie w stosunku do rozwiązań opartych o niezależny blok pamięci.

Operacje multipleksora są wykonywane przez odpowiednie mikroprogramy, przechowywane w tej samej pamięci stałej, co mikroprogramy procesora.

Kanały selektorowe mają wbudowane układowe rejestry sterujące operacjami WE-WY, a także korzystają z mikroprogramów zawartych w pamięci stałej. Szerokie zastosowanie sterowania mikroprogramowego w kanałach zmniejszyło ich logikę układową, zachowując jednocześnie wysoką przepustowość. Szybkość przesyłania kanału selektorowego wynosi 1100 Kbajtów/s. Natomiast sumaryczna prze-

Tabela 1. Jednostki Centralne JS EMC i ich główne parametry

Model		Szybkość procesora w oper/s	Pojemność pamięci operacyjnej (w Kb)	Ilość i szybkość kanałów (Kb)	
nazwa	szyfr			multiplekserowego	selektorowych
R-10	JS-1010	10000	8÷64	30	1/200
R-20	JS-1020	9000	64÷256	16	2/200
R-20A	JS-1021	40000	64÷256	35	2/250
R-30	JS-1030	55000	128÷512	40	3/800
R-32	JS-1032	200000	128÷1024	110	3/1100
R-40	JS-1040	320000	256÷1024	110	6/1300
R-50	JS-1050	500000	256÷1024	110	6/1300

Uwaga 1. Szybkości EMC od R-20 do R-50 są wzajemnie porównywalne (metodą Gibson I), gdyż stosują identyczne formaty informacji oraz jednakową listę instrukcji użytkowych natomiast R-10 ma odmienną i prostą listę rozkazów (właściwą dla minikomputerów).

Uwaga 2. Szybkość kanału multiplekserowego podana została dla pracy w trybie jednoczesnej obsługi wielu urządzeń zewnętrznych.

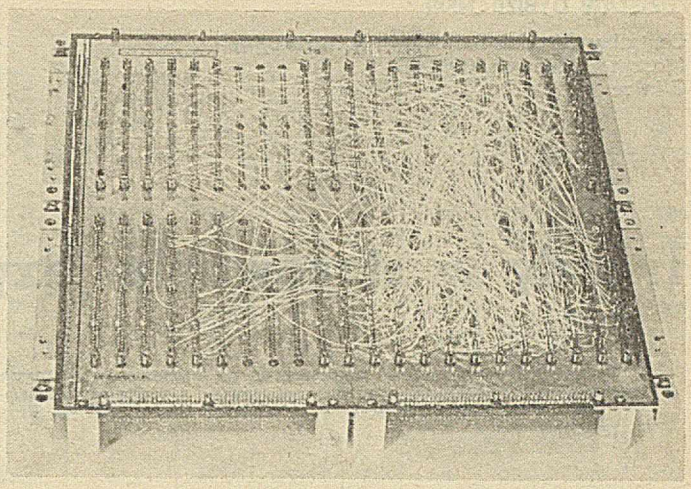


Foto 4. Tylna część kasety procesora

pustowość wszystkich kanałów wynosi 2,6 Mbajtów/s, podczas gdy teoretycznie maksymalna przepustowość — przy rozwiązaniu całkowicie układowym — mogłaby wynosić 3,3 Mbajtów/s. Podrożyłoby to jednak koszty kanałów w sposób nieproporcjonalny do uzyskanych efektów.

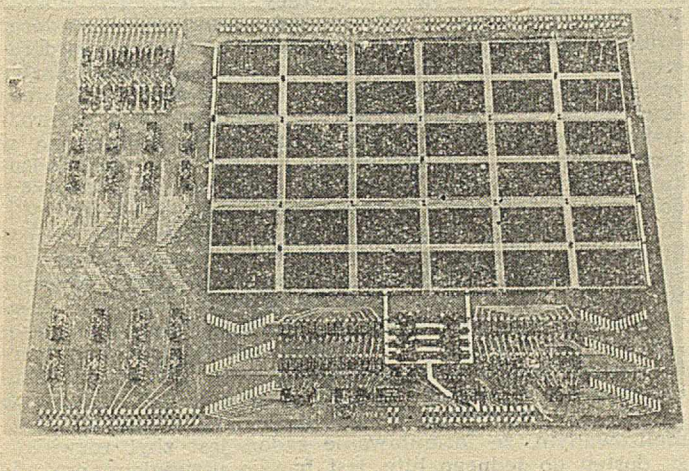


Foto 3. Moduł pamięci operacyjnej (16 Kb)

System kontroli i diagnostyki

W celu podniesienia niezawodności maszyny zastosowano — oprócz klasycznej kontroli nieparzystości — specjalne układy dynamicznej kontroli i zapamiętywanie stanu maszyny (erratografia), w którym znajdowała się ona w momencie pojawienia się błędu. Przy opracowywaniu rozwiązań maszyny R-32 kierowano się dwoma podstawowymi wymaganiami:

— średni użyteczny czas pracy — nie mniejszy niż 95%
— średni czas wznowienia pracy maszyny — nie dłuższy niż 30 minut.

W celu osiągnięcia powyższych wymagań wbudowano różne układy kontroli oraz układy diagnostyczne.

Przyjęty system kontroli obejmuje wiele punków przepływu informacji, co umożliwia ciągłe sprawdzanie poprawności pracy maszyny. Wykrycie błędu może spowodować wykonanie erratografii stanu maszyny, przerywanie programu i uaktywnienie diagnostycznej części systemu operacyjnego. Jeżeli uszkodzenie ma charakter chwilowy, to fakt wystąpienia błędu i wykonanie erratografii maszyny podczas jego zaistnienia mogą być wykorzystane do tworzenia statystki uszkodzeń maszyny. Natomiast w przypadku uszkodzenia stałego, wykonanie szeregu programowych procedur diagnostycznych pozwala na lokalizację uszkodzonego elementu.

Oprócz ciągłej kontroli wprowadzono również tzw. okresową kontrolę maszyny pozwalającą z określoną częstotliwością (co 20 ms) sprawdzić dokładnie arytometr za pomocą testów mikroprogramowych. Częstość i czas trwania tego testu są tak dobrane, aby nie zabierać więcej niż 1% czasu pracy maszyny.

W procesorze kontrolowane są w sposób ciągły następujące węzły przepływu informacji:

— układy interfejsu PAO-Procesor wraz z pamięcią operacyjną
— informacja czytana z Pamięci Mikroprogramów (PAM)
— układy wybierania mikroprogramów kanałowych
— układy interfejsu Kanał-Jednostki Sterujące urządzeniami wejścia—wyjścia.

Pamięć operacyjna ma wbudowane układy, umożliwiające autonomiczne sprawdzanie fizycznego bloku pamięci, za pomocą specjalnego testera i po uprzednim odłączeniu logicznym i fizycznym bloku od szyny PAO-Procesor. Pozostałe bloki mogą pracować w systemie po dokonaniu rekonfiguracji przy pomocy odpowiednich przełączników. Poza tym procesor ma w PAM zespół testów, sprawdzających wszystkie bloki PAO dowolną informacją zadaną z klawiatury pulpitu technicznego, tzw. trudnym szablonem pamięci. Istniejąca sygnalizacja na pulpicie technicznym pozwala na dokładną lokalizację błędu.

Ponadto w procesorze wbudowano specjalne rejestry z sygnalizacją, w których notuje się pojawienie się błędów w różnych układach. W celu uczynienia sprzętowych rozwiązań maszyny bardziej „diagnozowalnymi", wprowadzono nielegalną instrukcję DIAGNOZUJ, która powoduje uruchomienie określonej sprzętowej procedury diagnostycznej (np. zapamiętanie stanu rejestrów procesora, stanu kanału, stanu pamięci lokalnej, pamięci podkanałów, itp.).

ZDZISŁAW DYBIZBAŃSKI
Centralny Ośrodek Projektowo-Konstrukcyjny
Maszyn Górniczych KOMAG
Gliwice

Trendy w stosowaniu języków programowania*)

W artykule omówiono popularność w stosowaniu różnych języków programowania na podstawie materiałów zebranych w ośrodku obliczeniowym OUTOKUMPU OY w Finlandii.

Treść artykułu prawie w całości opiera się na opiniach analityków i programistów Ośrodka Obliczeniowego fińskiego koncernu OUTOKUMPU OY oraz na zebranych przez nich materiałach. Z tej przyczyny potrzebna będzie na wstępie krótka chociażby charakterystyka tego ośrodka, tym bardziej, że przedstawione w niej liczby zainteresują — jak sądzę — polskich informatyków.

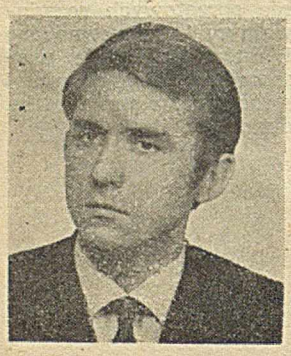
OUTOKUMPU OY jest największym koncernem górniczym w Finlandii, skupia 20 zakładów przemysłowych i instytutów, zatrudniając łącznie 10 tys. pracowników. Ośrodek obliczeniowy koncernu wyposażony jest w komputer IBM-360/40 z pamięcią operacyjną 192 K bajtów, z 12 jednostkami pamięci dyskowej dużej pojemności i 4 jednostkami pamięci taśmowej. Niezależnie każdy zakład wyposażony jest w minikomputer lub terminal. Ośrodek istnieje od 12 lat, a wymieniony komputer eksploatowany jest od 7 lat. W okresie stażu autora przygotowywano się do wymiany jednostki centralnej na IBM-370/145 oraz szeregu urządzeń dodatkowych.

Powierzchnia użytkowa pomieszczeń ośrodka wynosi 2000 m², z czego sala EMC zajmuje aktualnie 160 m². Bezpośrednio w ośrodku zatrudnionych jest 26 pracowników koncepcyjnych, 35 pracowników w dziale eksploatacji EMC oraz 17 pracowników administracyjnych i pomocniczych.

Komputer pracuje 24 godziny na dobę nie wyłączając sobót i niedziel, a pełne jego obciążenie wynika z eksploatacji 4 dobrze działających, dużych systemów:
— „Produkcja — Planowanie i Rozliczanie" (45% czasu)
— „Gospodarka Materiałowo-Magazynowa" (20%)
— „Księgowość" (20%)
— „Kadry i Płace" (20%)

Pozostałe 5% przypada na zastosowania inżynieryjno-techniczne, które są głównie realizowane za pomocą dzierżawionego terminala.

*) Artykuł został napisany w oparciu o konsultacje oraz materiały zebrane w okresie 3-miesięcznego stażu autora (I kwartał 1975 r.) w Ośrodku Obliczeniowym koncernu OUTOKUMPU OY w Finlandii.



Mgr inż. ZDZISŁAW DYBIZBAŃSKI ukończył studia na Wydziale Górniczym Politechniki Śląskiej w 1969 r. Po rocznej pracy „w ruchu" w kopalni „Rozbark" przeszedł do pracy w zakresie projektowania systemów EPD w branży maszyn górniczych. W pracy zawodowej i naukowej zajmuje się głównie uogólnionymi systemami przetwarzania danych. Obecnie pracuje w Centralnym Ośrodku Projektowo-Konstrukcyjnym Maszyn Górniczych „KOMAG" w Gliwicach na stanowisku zastępcy kierownika Branżowego Ośrodka Maszyn Matematycznych. Współpracuje z Instytutem Organizacji i Ekonomiki Górnictwa Wydziału Górniczego Politechniki Śląskiej.

Głównym tematem stażu w Ośrodku Obliczeniowym OUTOKUMPU OY były zagadnienia związane z budową programów uogólnionych, a trendy w stosowaniu języków programowania rozpoznałem „ponadplanowo" sądząc, że doświadczenia i opinie fińskie są tego warte.

* * *

Ośrodek obliczeniowy w OUTOKUMPU OY posiadał w lutym 1975 roku ok. 3000 własnych modułów programów, z których 64% napisano w ASSEMBLERZE, 20% w PL/1, a 16% w FORTRANIE.

Jest charakterystyczne, że zarówno w tym przedsiębiorstwie jak i w innych przedsiębiorstwach fińskich ewidencjonuje się moduły programów, nie zaś kompletne programy. Wynika to z rozsądnych decyzji kierownictw ośrodków zobowiązujących programistów do programowania wyłącznie w sposób modułowy. Do stosowania tego sposobu zachęcają zresztą możliwości systemów operacyjnych IBM i biblioteka modułów firmowych. I tak np. opis struktury zbioru oraz struktury zapisu (rekordu) nie jest umieszczony w programie, pisze się natomiast rozkaz przywołujący oraz nazwy opisów struktur w bibliotece. Jest to oczywiście duża oszczędność w pracochłonności programowania, a co ważniejsze znacznie ułatwia wprowadzanie zmian zarówno w programach jak i w strukturze rekordu lub zbioru. Dokumentacja modułów programów jest sporządzana bardzo dokładnie i starannie ewidencjonowana.

W trakcie licznych rozmów, jakie przeprowadziłem z analitykami i programistami, starałem się wysondować ich opinię na temat poszczególnych języków programowania. W OUTOKUMPU OY COBOL nie jest używany i nie przewiduje się jego wprowadzenia. Tamtejsi analitycy i programiści nie twierdzą bynajmniej że COBOL jest językiem złym, jednak w ich przypadku, w przypadku przedsiębiorstwa przemysłowego jest on znacznie mniej użytecznym aniżeli PL/1. Twierdzą, że COBOL jest językiem zbyt sformalizowanym, że formułowanie zadań jest zbyt rozwlekłe, że dostęp do jednego bitu jest bardzo utrudniony oraz, że nie jest możliwe jego użycie do napisania programów uogólnionych zdolnych interpretować specyfikowane parametry np. w postaci matematycznego zapisu funkcji arytmetycznej. Możliwości te daje natomiast PL/1, który ma jeszcze takie zalety jak: łatwość sporządzania dokumentacji i fakt, że wystarcza tylko częściowa znajomość języka w przypadku programowania dla problemów w określonym zakresie. Wielorakie możliwości funkcjonalne języka PL/1 należy — jak sądzę — uznać za szczególnie ważne wobec rysujących się tendencji opracowywania programów uogólnionych zdolnych do realizacji różnorodnych, niemożliwych do uprzedniego przewidzenia potrzeb użytkowników.

W OUTOKUMPU OY zadecydowano już, że wszystkie nowe programy zarówno dla systemów EPD jak i dla obliczeń inżynieryjnych będą pisane w PL/1. Nie wyklucza to oczywiście możliwości użycia FORTRANU czy ASSEMBLERA przez znających te języki inżynierów spoza ośrodka.

* * *

Aby jednak nie budować opinii o tendencjach w zakresie stosowania języków programowania na bazie doświadczeń jednego ośrodka obliczeniowego, sięgnijmy do specjalnych fińskich opracowań na ten temat.

Jedno z nich [1] zostało opublikowane w 1973 r. w Helsinkach przez komitet rządowy powołany do przeanalizowania istniejącej sytuacji w zakresie zastosowań komputerów w

Tabela Ia. Stosowane języki programowania w przedsiębiorstwach fińskich

Instytucja	Liczba modułów programowych	% z ogólnej liczby modułów	Języki						
			ASSEMBLER	FORTRAN	ALGOL	COBOL	FAS	RPG	PL/1
Przedsiębiorstwa rządowe									
Centrum Obliczeniowe Rządu	2140	21,5	1600	50	—	20	250	—	220
Fińskie Koleje Państwowe	1215	12,2	345	50	—	—	—	20	800
Państwowy Centralny Urząd Statyst.	1200	12,0	150	20	—	—	—	800	230
Obrona Narodowa	663	6,7	330	263	—	70	—	—	—
Biuro Projektów Mostów i Dróg	505	5,1	125	69	4	—	1	—	306
Poczta i Telegraf	445	4,6	307	25	—	2	30	—	91
Departament Leśnictwa	220	2,2	120	10	—	40	—	50	—
Urząd Podatkowy	110	1,1	90	—	—	—	20	—	—
Główny Urząd Geodezyjny	105	1,1	10	95	—	—	—	—	—
Ministerstwo Szkolnictwa	1	0,0	—	1	—	—	—	—	—
Liczba modułów razem	6614	66,4	3077	583	4	132	301	870	1647
Procentowy udział języków	100%	—	46,5%	8,8%	0,0%	2,0%	4,6%	13,2%	24,9%

Tabela Ib. Stosowane języki programowania w fińskich ośrodkach naukowych

Instytucja	Liczba modułów programowych	% z ogólnej liczby modułów	Języki						
			ASSEMBLER	FORTRAN	ALGOL	COBOL	FAS	RPG	PL/1
Szkoły Wyższe i Instytuty Badawcze									
Państwowe Centrum Badawcze Techn.	1215	12,2%	5	1100	100	10	—	—	—
Wyższa Szkoła Techn. w Helsinkach	710	7,1%	30	530	150	—	—	—	—
Instytut Meteorologiczny	609	6,1%	89	—	520	—	—	—	—
Uniwersytet w Helsinkach:									
— Dział Przetwarzania Danych	244	2,5%	—	102	82	60	—	—	—
— Fizyka Atomowa	22	0,2%	2	20	—	—	—	—	—
— Chemia	15	0,2%	—	3	12	—	—	—	—
Wyższa Szkoła Techn. w Tampere	210	2,1%	50	180	—	—	—	—	—
Uniwersytet w IYVASKYLA	100	1,0%	—	100	—	—	—	—	—
Uniwersytet w Oulu	95	1,0%	—	90	5	—	—	—	—
Instytut Badawczy Rolnictwa	70	0,7%	10	60	—	—	—	—	—
Instytut Badań Oceanograficznych	55	0,6%	—	—	55	—	—	—	—
Liczba modułów razem	3345	33,6%	186	2165	924	70	—	—	—
Procentowy udział języków	100%	—	5,6%	64,7%	27,6%	2,1%	—	—	—
Ogólna liczba modułów (tab. 1a + 1b)	9959	100%	3263	2748	928	202	301	870	1647
Procentowy udział języków (ogóln.)	100%	—	32,8%	27,6%	9,3%	2,0%	3,0%	8,7%	16,5%

urzędach państwowych, w państwowych instytutach naukowo-badawczych oraz w szkołach wyższych. Jeden z rozdziałów tego opracowania poświęcony jest językom programowania i zawiera tabelę obrazującą udział poszczególnych języków w istniejących programach (Tab. 1a i 1b). Z tabel tych wynika wyraźnie, że w przedsiębiorstwach i urzędach na pierwszym miejscu znajduje się ASSEMBLER na drugim zaś PL/1. Znamienny jest tu bardzo niski procent programów w COBOLU. Stosunkowo wysoki procent programów w języku RPG (do programowania wydruków zestawień tabelarycznych) wynika jedynie z faktu, że zastosował go Centralny Urząd Statystyczny. ALGOL nie jest właściwie wcale stosowany.

W szkołach wyższych i instytutach badawczych natomiast zdecydowane pierwszeństwo należy do FORTRANU a ALGOL plasuje się daleko za nim. Zaś ASSEMBLER i COBOL z uwagi na specyfikę tych instytucji stanowią margines zastosowań, FAS, RPG i PL/1 nie są w ogóle stosowane.

*

* *

Z innej publikacji [2] wynika, że problem „COBOL czy PL/1" nurtuje wiele ośrodków obliczeniowych w Finlandii. Istniejące tam stowarzyszenie użytkowników dużych komputerów podjęło się trudnego zadania porównania stopnia przydatności obu tych języków.

Wynikiem pracy specjalnie powołanego zespołu nie było bynajmniej zawyrokowanie o wyższości któregoś z obu języków lecz metoda punktowa umożliwiająca każdemu ośrodkowi obliczeniowemu dokonanie takiego porównania. Wyniki porównania zależą bowiem od faktycznej skali specyficznych potrzeb poszczególnych użytkowników języka. Raport zespołu został opublikowany w 1974 r. Z raportu tego przytaczam tabelę obrazującą procentowy udział poszczególnych języków w programach opracowanych przez wybrane ośrodki obliczeniowe (tab. II).

Tabela ta dostarcza również informacji na temat typów stosowanych komputerów, wielkości pamięci operacyjnej, zastosowanych systemów operacyjnych oraz ogólnej liczby modułów programów.

Pomimo, że w tabeli tej umieszczono wyłącznie czołowych obecnie użytkowników COBOLU lub PL/1 wyraźnie rysuje się **dominująca jeszcze rola ASSEMBLERA.** Wynika to z następujących przyczyn:

— analiza objęła wszystkie użytkowane moduły programów, a więc także cały szereg programów napisanych bezpośred-

Tabela II. Tabela porównawcza stosowanych języków PL/1 oraz COBOL u czołowych użytkowników w Finlandii

	Przedsiębiorstwo	Komputer	Pojemność PAO Kbajtów	System operacyjny	Ogólna liczba modułów progr.	Procentowy udział języków w ogólnej liczbie modułów programów								
						COBOL %	ASSEM. %	PL/1 %	FOR-TRAN %	RPG %	FAS %	RPG II %	BASIC %	ALGOL %
Dominacja PL/1 nad COBOL-em	VR (koleje państwowe)	IBM 370/145	384	OS/R 21,7	2700	—	60,8	36,3	2,2	0,7	—	—	—	—
	Pohjola (przedsiębiorstwo ubezpieczeniowe)	IBM 370/145	512	OS/VS1	1200	0,5	94,5	5,0	0,5	—	—	—	—	—
	Outokumpu (górnictwo i hutnictwo metali kolorowych)	IBM 360/40	192	OS/MFT II	2472	—	66,0	16,0	18,0	—	—	—	—	—
	SOK (duża organizacja spółdzielcza w zakresie produkcji i sprzedaży)	IBM 370/155	384	OS/MFT II	1100	1,0	35,0	63,0	—	1,0	—	—	—	—
	Neste (fińskie przedsiębiorstwo naftowe)	IBM 370/145	256	OS/VS1 2,0	844	—	43,2	37,0	12,2	7,6	—	—	—	—
	TKP (komputerowe przedsiębiorstwo usługowe)	IBM 370/155	512	OS/MFT—HASP	pow. 3000	—	95,0	5,0	—	—	—	—	—	—
	OTK (duża organizacja spółdzielcza)	IBM 370/155	512	OS/MVT,DOS	700	2,9	57,1	10,0	21,4	—	8,6	—	—	—
Dominacja COBOL-u nad PL/1	Tietotehdas (komputerowe przedsiębiorstwo usługowe)	IBM 370/155	768	OS/R 21,6	4000	50,0	45,0	—	5,0	—	—	—	1,0	—
		IBM 370/145	512	OS/R 21,6										
	OKO (organizacja miejszych banków)	IBM 370/145	384	OS/MFT II	1700	79,0	18,0	—	3,2	—	—	—	—	—
	Enso-Gutzeit (przeds. przem. drzewno-papiern.)	IBM 360/25	48	DOS	1000	92,0	2,0	—	6,0	—	—	—	—	—
		IBM 360/40	128	DOS										
	Sanoma Oy (przedsięb. wydawn. periodycznych)	IBM 360/40	192	DOS	360	40,0	60,0	—	—	—	—	—	—	—
		IBM 360/20	8	KORTTI + TOS										
	Postipankki (bank państw.)	3×370/145	3×256	DOS										
		1×370/135	1×144	DOS	3500	15,0	80,0	—	—	3,0	—	2,0	—	—
		2×360/30	2×64	DOS										
	Hankkija (rolnicza organizacja spółdzielcza)	2×GE-400	16K słów	ATPS	brak danych	90,0	10,0	—	—	—	—	—	—	—
		1×HGO-40	96K słów	GEOS										
	Kela (państw. biuro ubezpieczeń społecznych)	370/155	512	OS/MKT II	1500	1,0	94,0	1,0	4,0	—	—	—	—	—
		370/145	384	OS/MKT II										
	Średni procent					26,5	54,2	12,0	5,2	1,0	0,7	0,2	0,2	0,1

nio po wprowadzeniu na tutejszy rynek, w końcu lat 60-tych, komputerów IBM serii 360,

— programistom znającym języki symboliczne, stosowane w poprzednich typach maszyn, najłatwiej było przestawić się na podobny do nich ASSEMBLER mimo, że obiektywnie jest to język trudniejszy do nauczania się aniżeli np. FORTRAN

— możliwości funkcjonalne ASSEMBLERA są znaczne

— przy masowym przetwarzaniu nie bez znaczenia były: zajmowanie stosunkowo małego obszaru pamięci operacyjnej oraz szybki czas realizacji programu

— pierwsze kompilatory PL/1 nie były jeszcze dostatecznie sprawdzone a kompilowane programy nie mieściły się często w pamięci ferrytowej maszyn serii IBM 360

— do stosowania COBOLU wiele ośrodków nie było przygotowanych.

Wg zsumowanych procentów w tabeli II na **drugim miejscu plasuje się COBOL.** Wynika to zapewne z faktu, że użytkownicy, którzy zdecydowali się na wprowadzenie COBOLU musieli przygotować wiele modułów w tym języku, już chociażby dlatego, że język nie nadaje się do przygotowania programów uogólnionych. Tezę tę można poprzeć następującym faktem: w ośrodku obliczeniowym OUTOKUMPU OY dla każdego z 5 makro-systemów napisano kilka lub przynajmniej 1 program uogólniony w języku ASSEMBLER a programy te są wykrzystywane dla realizacji różnorodnych potrzeb użytkowników ośrodka. W większości przypadków w COBOLU musianoby napisać odrębne moduły dla każdego zapotrzebowania, a zatem ich liczba musiałaby znacznie wzrosnąć. Pozostaje jednak faktem, że większość ośrodków, które wprowadziły u siebie COBOL nie sięga do PL/1.

Na trzecim miejscu znalazł się PL/1, ale jest to niewątpliwie język, który będzie odgrywał znacznie większą rolę w przyszłości. Specjaliści, z którymi dyskutowałem to zagadnienie są zdania (a nie jest to chyba pochopna opinia bo na jej podstawie podejmuje się już teraz konkretne decyzje), że ośrodki, które zdążyły już COBOL „rozkręcić" u siebie, pozostaną przy nim, pozostałe natomiast zaprzestaną pisania programów w języku ASSEMBLER i przejdą na PL/1, tym bardziej, że język ten może również w pełni zastąpić FORTRAN i — jeśli ktoś używał — ALGOL. Głównym sprawcą nagłego wzrostu zainteresowania językiem PL/1 jest zapewne pamięć wirtualna zastosowana w komputerach IBM serii 370, które już wkrótce prawie w pełni zastąpią swoich poprzednich z serii 360.

Jak już wspomniałem wcześniej, przetłumaczony program w PL/1 zajmuje stosunkowo dużo miejsca a właśnie pamięć wirtualna umożliwia pominięcie tego uwarunkowania. Należy podkreślić, że stosowanie pamięci wirtualnej w żaden sposób nie obciąża ani programisty, ani operatora. Ponadto ostatnio dostępna jest już zoptymalizowana wersja kompilatora PL/1, który nie powoduje już tak dużej zajętości pamięci. (Tę właśnie wersję zakupi ośrodek OUTOKUMPU OY). Tak więc od momentu wprowadzenia pamięci wirtualnej należy się chyba spodziewać szybkiego wzrostu zastosowań języka PL/1.

Język FORTRAN w tabeli II znajduje się na czwartym miejscu i osiągnął nawet znaczny procent zastosowań zważywszy, że charakter analizowanych przedsiębiorstw implikuje typowe przetwarzanie danych. Natomiast w zastosowaniach naukowych i inżynierskich jego pierwsza pozycja wydaje się nie zachwiana.

Język RPG opracowany przez IBM miał zaspokoić konkretne zapotrzebowanie ośrodków obliczeniowych na szybkie i łatwe programowanie wydruków z istniejących zbiorów. Niestety, jak wynika z przeprowadzonych rozmów, język ten nie spełnia pokładanych w nim nadziei. Jest zbyt skomplikowany, zbyt trudny do nauczenia a czas programowania nawet przy dużej wprawie nie jest wystarczająco krótki. W rezultacie ośrodki szukają innych dróg do ułatwienia i usprawnienia swojej pracy.

Wobec niepowodzenia jakie spotkało specjalistyczny język programowania wydruków, ośrodkom obliczeniowym (i to chyba nie tylko w Finlandii) pozostały obecnie do wyboru dwie drogi: albo zakupić uogólnione programy sterowane przy pomocy zadawanych parametrów albo też opracować takie programy we własnym zakresie. Jest to jednak temat wykraczający poza ramy artykułu, niemniej wymagający pilnej dyskusji.

PAS jest językiem opracowanym w fińskim Centrum Obliczeniowym Rządu. Zasadniczą przyczyną jego opracowania była chęć pełnego uniezależnienia się od typów komputerów — dla każdego typu komputera mogą sami przygotować kompilator. Możliwości funkcjonalne tego języka można scharakteryzować jako pośrednie między ASSEMBLEREM a FORTRANEM. Język ten jest łatwy do nauczenia, posiada jednak dużo ograniczeń formalnych. Jak wynika z tabeli I i II oprócz Centrum Rządowego stosują go także inne ośrodki obliczeniowe.

Język BASIC stosowany jest głównie do obliczeń inżynierskich wykonywanych przy użyciu dzierżawionych terminali i stąd w tabeli II wykazany procent stosowania jest tak nikły.

ALGOL w fińskich przedsiębiorstwach nie jest używany. Jak już wspomniałem przy omawianiu tabeli I stosuje się go w niektórych uczelniach i instytucjach badawczych, jednakże i tam, w zakresie obliczeń naukowo-technicznych dominuje zdecydowanie FORTRAN.

Wnioski

1. Przygotowując kadrę programistów dla komputerów JS EMC korzystne będzie już obecnie podjęcie decyzji wyboru najbardziej odpowiedniego w danych warunkach języka programowania.

2. Równoczesne stosowanie w ośrodku obliczeniowym COBOL-u i PL/1 będzie najczęściej niecelowe.

3. Język PL/1 jest zdecydowanie godny polecenia dla ośrodków obliczeniowych które:
— nie są zbytnio zaangażowane w COBOLU-u lub
— planują opracowywanie programów uogólnionych lub
— planują wykorzystywać komputer zarówno do zastosowań systemowych jak i naukowo-technicznych.

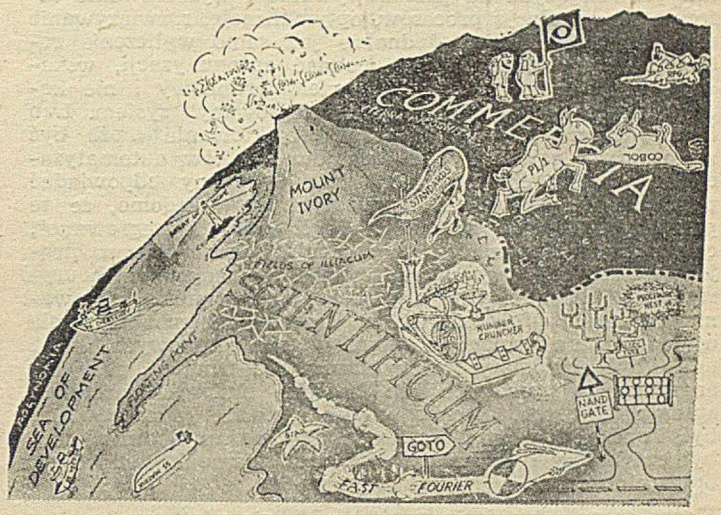
4. Ewentualną decyzję wprowadzenia języka RPG poprzedzić dokładnym rozeznaniem jego możliwości funkcjonalnych oraz skali trudności w stosowaniu.

5. Niezależnie od szkolenia programistów dla komputerów RIAD niezmiernie istotne będzie dobre przygotowanie specjalistów w zakresie systemów operacyjnych — bardzo wskazane m.in. staże u zaawansowanych użytkowników IBM/360.

6. Celowe wydaje się wprowadzanie w Ośrodkach obliczeniowych zasady oprogramowywania systemów informatycznych za pomocą modułów programowych, co okaże się szczególnie korzystne przy stosowaniu systemów operacyjnych JS EMC.

Bibliografia

[1] Voltion ATK: N Konferiippumattomuus-Toimikunnanmietinto, Helsinki 1973
[2] Tutkimusraporetti COBOL — PL/1 Kielivertailu, Helsinki 1974



WŁODZIMIERZ TRAFALSKI
Centrum Techniki Okrętowej
Gdańsk

Elementy strategii automatyzacji prac projektowych

Omówiono elementy strategii automatyzacji prac projektowych z punktu widzenia użytkownika-projektanta oraz użyteczności działań automatyzujących w stosunku do oceny wartości samego programu. W artykule podjęto próbę wyodrębnienia i opisu czynników mających zasadnicze znaczenie przy automatyzacji prac projektowych.

Świadomość faktu, że na pewnym, bardziej zaawansowanym etapie rozwojowym proces automatyzacji prac projektowych może wejść w kolizję z procesem projektowania jest coraz powszechniejsza. Nie jest, ostatecznie, rzeczą zbyt trudną takie zaprojektowanie i zrealizowanie układów automatyzacji prac, aby proces projektowania został całkowicie sparaliżowany a nadrzędzie obróciło się przeciw swojemu użytkownikowi — są smutne przykłady z praktyki, gdzie taki stan został osiągnięty znacznym wysiłkiem ludzi i sporym nakładem kosztów. Zjawisko takie jest sygnałem, że określona sytuacja problemowa, zmuszająca człowieka do działań automatyzacyjnych nie została dostatecznie rozpoznana i właściwie sformułowana. Wspomaganie procesu projektowania przez automatyzację pewnych ogniw tego procesu wymaga — wydaje się — specyficznego spojrzenia na styk obu procesów: automatyzacji i projektowania. Nie ulega przy tym wątpliwości, że powinno to być spojrzenie z perspektywy użytkownika-projektanta, którego interesują wyłącznie użytkowe cechy proponowanych mu sposobów automatyzacji. W odczuciu autora istnieje wyraźna nadrzędność tego poziomu oceny użyteczności działań automatyzujących w stosunku do poziomu oceny samego programu „jako takiego", czym się zajmuje tak zwana inżynieria oprogramowania. Może się bowiem zdarzyć — co nie jest żadnym odkryciem — że bardzo elegancko rozwiązany program nie spełni pokładanych w nim nadziei, jeżeli zostaną źle odczytane rzeczywiste potrzeby użytkownika.

Mgr inż. WŁODZIMIERZ TRAFALSKI ukończył Wydział Budowy Okrętów Politechniki Gdańskiej w 1957 r., a następnie przez 4 lata pracował w Biurze Konstrukcyjnym Stoczni Gdańskiej w dziedzinie obliczeń projektowych. Od 1961 r. zajmował się zagadnieniami konstrukcji kadłuba, a od 1975 r. pełni funkcję Kierownika Ośrodka Mechaniki Konstrukcji Okrętu. Jest autorem szeregu algorytmów i opracowań metodycznych z zakresu konstruowania oraz automatyzacji prac inżynierskich.

Dobroć programu — jeżeli można użyć takiego określenia, obejmującego wszystkie pozytywne cechy dobrego programu — jest oczywiście jednym z walorów jego użyteczności, ale istnieje szereg innych czynników równie ważnych. Podjęta próba ich wyodrębnienia i opisu dokonana została na tle prac związanych z projektowaniem automatyzacji działań w procesie projektowania okrętu, jednakże wnioski wydają się na tyle ogólne, że mogą zainteresować użytkowników z innych dziedzin techniki.

Poziomy automatyzacji

Automatyzacja prac projektowych rozumiana jest na ogół jednopoziomowo — jako działanie prowadzące do stworzenia systemu obliczeniowego, czy wspomagającego projektowanie w którym nie tylko zrealizowane zostaną wszystkie czynności związane z projektowaniem a poddające się automatyzacji, ale ponadto — spotęgowane zostaną możliwości twórcze projektanta dzięki wykorzystaniu specjalnych technik pozwalających na przechowywanie i wyszukiwanie informacji, symulację procesów, optymalizację, rozwiązywanie równań, kreślenie itp. Jednakże już samo wyliczenie rodzajów pracy, jakie można poddać automatyzacji, wskazuje, że proces ten może być wielokierunkowy i nie koniecznie nastawiony na jeden „zintegrowany" system. Lub inaczej mówiąc — w procesie projektowania może być zastosowana pewna liczba rodzajów środków automatyzacji o określonych funkcjach, które powinny odpowiadać potrzebom użytkownika. Z praktyki zaś wiadomo, że te potrzeby są bardzo różne i na ogół mało „zintegrowane", co jest wynikiem nie tyle nieuporządkowania procesu projektowania, ale jego właściwością, wynikającą z heurystycznego charakteru tego procesu. Każda próba ujmowania czynności projektowych w logiczny schemat działania ujawnia różne poziomy i zakresy prac rutynowych, które mogą być zautomatyzowane łatwiej lub trudniej, przy zaangażowaniu prostszych lub bardziej złożonych urządzeń automatyzujących. Zachodzi więc pytanie, jaka powinna być konfiguracja tych środków, aby proces automatyzacji był optymalny i co należy uznać za główne kryterium tej optymalności?

Konsekwencją przyjęcia punktu widzenia użytkownika-projektanta na proces automatyzacji prowadzonych przez niego prac, jest wybór takiego celu optymalizacji, który będzie zmierzał do racjonalizacji wysiłku i zmniejszenia pracochłonności, a więc i kosztu pracy. Badania na temat, przeprowadzone przez R.E. Millera i J. McCaslina [1] na materiale zebranym z amerykańskich biur projektowych i reprezentatywnym dla 3000 takich organizacji doprowadziły do bardzo ciekawych wniosków. Autorzy przeprowadzili inwentaryzację prac inżynierskich oraz dokonali szczegółowej charakterystyki (24 parametry) każdego rodzaju pracy (143 rodzaje prac), a także — opisali parametry pracy siedmiu rodzajów urządzeń automatyzujących (suwak rachunkowy, kalkulator mechaniczny, kalkulator programowy, mały system wieloprocesorowy, mała maszyna wsadowa, duża maszyna wsadowa), budując program symulujący pracę tych urządzeń nad realizacją określonych zadań inżynierskich. Zakres analizy obejmował poza wyborem i opisem typów prac i urządzeń automatyzujących:

— analizę kosztu wykonania każdej pracy za pomocą każdego urządzenia automatyzującego dla wyznaczenia urządzenia o najmniejszym koszcie

— bilans całkowitego kosztu pracy inżynierskiej w relacji do kosztu urządzeń automatyzujących dla różnych konfiguracji tych urządzeń.

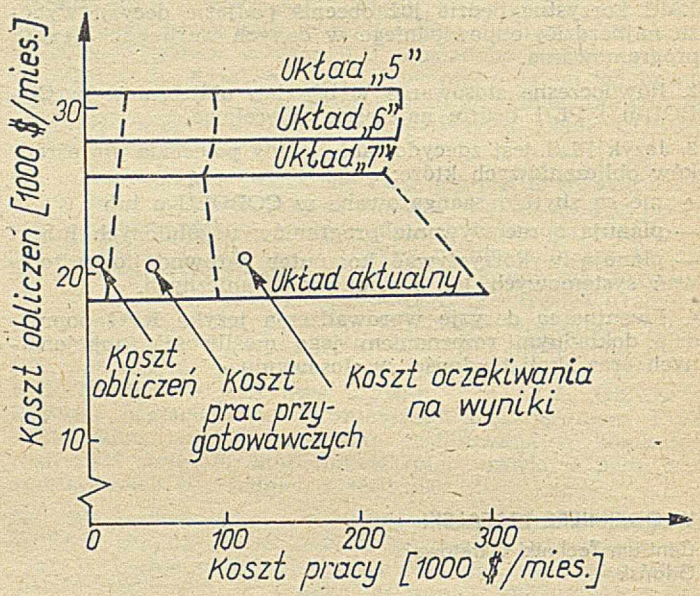
W koszcie pracy ujęto wszystkie czynności i składniki związane z jej realizacją, począwszy od pisemnego sformułowania zadania i skompletowania danych, co obejmowało:

1. pracochłonność przygotowania danych
2. pracochłonność realizacji obliczeń
3. pracochłonność wyprowadzenia (uporządkowania) wszystkich wyników
4. koszt nieproduktywnego czasu oczekiwania na wyniki
5. koszt pracy maszyny
6. koszt przygotowania nośników informacji.

Wynikający z powyższej analizy optymalny procentowy rozdział prac na poszczególne typy urządzeń automatyzacyjnych pokazano w tabeli I. W tabeli tej „układ aktualny" obrazuje sytuację zarejestrowaną w momencie przeprowadzania inwentaryzacji prac (rok 1968), natomiast układ „7", „6" i „5" obrazuje symulowane zestawy urządzeń automatyzujących analizowane pod względem kosztu wykonania pracy.

Tabela I. Procentowy rozdział prac na poszczególne typy urządzeń automatyzujących, wg [1]

Rodzaj urządzenia	Układ aktualny	Układ „7"	Układ „6"	Układ „5"
Suwak rachunkowy	13	0	0	0
Kalkulator mechaniczny	41	15	15	15
Kalkulator programowy	6	33	36	36
Mały system wieloprocesowy	—	7	13	—
Duży system wieloprocesowy	—	10	22	35
Mała maszyna wsadowa	—	30	—	—
Duża maszyna wsadowa	40	5	14	14



Rys. 1. Koszt wykonania pracy w zależności od zestawu urządzeń automatyzujących

Wyniki analizy kosztowej dla powyższych konfiguracji urządzeń przedstawiono na rys. 1. Wynikają stąd następujące wnioski:

a) Układ aktualny jest nastawiony na minimalny koszt urządzeń automatyzujących, co jednak ma wyraźnie niekorzystny wpływ na całkowity koszt realizacji pracy. Jest to bezsprzecznie wynik praktycznej strategii działania ukierunkowanej na minimalizację kosztu urządzeń.

b) Pozostałe układy pracujące przy optymalnym rozdziale prac legitymują się wyższym kosztem urządzeń ale całkowity koszt pracy spada przy tym dość wyraźnie. Dla najbardziej optymalnego systemu „7" ten efekt osiąga się przez przerzucenie części prac:

— z urządzeń mechanizujących (suwak, kalkulator mechaniczny) na kalkulatory programowane

— z dużej maszyny wsadowej na mniejsze maszyny, łatwiej dostępne dla użytkownika.

c) Podstawowym składnikiem całkowitego kosztu pracy jest koszt oczekiwania na wyniki, stanowiący około 60% całości kosztów i ulegający największej redukcji w procesie optymalizacji (przyjęto przy tym, że połowa czasu oczekiwania może być wypełniona innymi pracami, co uwzględniono w bilansie).

Powyższe wnioski są bardzo ważne dla strategii automatyzacji prac projektowych. Ich konsekwencją jest przyjęcie następujących przesłanek:

I. Minimalizacja całkowitego kosztu wykonania pracy nie wyklucza nasycenia procesu projektowania urządzeniami automatyzującymi — usprawnienie działalności projektanta przez stworzenie mu szeregu możliwości automatyzacji prac jest równoznaczne z obniżeniem kosztu ich wykonania, przy czym sam koszt instalacji ma drugorzędne znaczenie.

II. Głównym parametrem warunkującym obniżenie kosztu wykonania pracy jest czas oczekiwania na wyniki w którym tkwią największe rezerwy oszczędności w świetle aktualnej praktyki, a zatem — system optymalny będzie się charakteryzować możliwie najmniejszym czasem oczekiwania, niemożliwym do zapełnienia inną pracą.

III. Podstawą racjonalnej strategii automatyzacji prac projektowych jest punkt wyjścia związany z analizą czasu oczekiwania na wyniki w której koszt i rodzaj urządzeń automatyzujących jest wynikowy a czas oczekiwania — najmniejszy. Studiując rozdział prac na poszczególne typy urządzeń automatyzujących, przedstawiony w tabeli I, można generalnie wyróżnić trzy grupy tych urządzeń, tworzące trzy główne poziomy automatyzacji:

— poziom podstawowy, reprezentowany przez „technikę stołową" (kalkulatory — głównie programowane), który, jak wynika z analizy, w każdym przypadku załatwia połowę wszystkich prac poddających się automatyzacji

— poziom średni, reprezentowany przez maszyny średniej wielkości (tu: mała maszyna wsadowa + mały system wieloprocesowy dla systemu „7"), realizujące pojedyńcze programy lub mniejsze pakiety programów, co dotyczy około 35% prac

— poziom wyższy, reprezentowany przez duże maszyny, realizujące pozostałe 15% prac (tu: duża maszyna wsadowa + duży system wieloprocesowy).

Uzyskuje się w ten sposób rozdział średniego obciążenia poszczególnych poziomów automatyzacji w relacji 50—35—15% prac, jako orientacyjny wskaźnik przy określaniu struktury planowanych urządzeń automatyzujących.

Układy automatyzacji

Wyznaczenie poziomów automatyzacji prac projektowych nie wyczerpuje problemu optymalnego doboru urządzeń automatyzujących, które muszą dobrze spełniać zadanie minimalizacji czasu oczekiwania na wyniki, zgodnie z wyprowadzonymi przesłankami. Przyjmując zatem czas oczekiwania na wyniki, jako główny parametr klasyfikacji prac, można wyróżnić następujące rodzaje działań, poddających się automatyzacji:

D1. obliczenia i prace wymiarujące, czysto projektowe, warunkujące powstanie koncepcji projektowej i tworzące główną procedurę projektowania w której czas oczekiwania projektanta nie da się z reguły zapełnić żadną inną pracą

D2. obliczenia i prace weryfikacyjne, sprawdzające koncepcję projektową już ukształtowaną i tworzące pomocnicze procedury projektowania w których czas oczekiwania projektanta lub specjalisty jest możliwy do częściowego zagospodarowania inną pracą

D3. obliczenia i prace dodatkowe uzupełniające, nie wpływające bezpośrednio na postać rozwiązania projektowego w których czas oczekiwania może być w całości zagospodarowany inną pracą.

Dodatkowymi czynnikami mającymi wpływ na dobór urządzeń automatyzujących są złożoność obliczeń i forma wyników.

Złożoność obliczeń jest czynnikiem określającym skalę działań obliczeniowych wymaganych lub pożądanych dla poprawnego rozwiązania problemu projektowego. Można tu wyróżnić także trzy podstawowe stopnie złożoności obliczeń:

Z1. modele matematyczne, funkcjonalne lub strukturalne, systemów (obiektów) o dużej liczbie zmiennych, formułowane z reguły jako modele optymalizacyjne

Z2. algorytmy obliczeniowe o stałej strukturze, będące z reguły fragmentami większych modeli matematycznych, ale realizowane jako samodzielne procedury

Z3. zadania obliczeniowe, z reguły o pomocniczym charakterze, realizowane raczej na podstawie wzorów niż algorytmów.

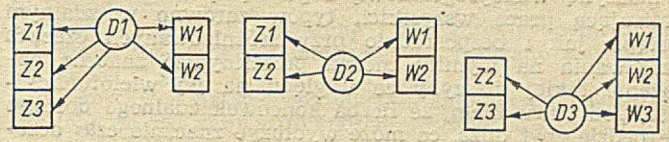
Wymagania w zakresie formy wyników obliczeniowych można uporządkować w następujących grupach:

W1. informacja alfanumeryczna, uporządkowana w odpowiednich tabelach i kompletach wyników

W2. informacja graficzna, opisana w stopniu umożliwiającym identyfikację wykresów lub rysunków

W3. informacja wyłącznie treściowa (słowna) — np. w systemach informacji źródłowej.

Każdy typ działania ma określone potrzeby realizacyjne w zakresie złożoności obliczeń i formy wyników, co z kolei przesądza o typie urządzeń. Potrzeby te mogą być różne w różnych dziedzinach projektowania technicznego, jednakże ich podstawowa struktura, pokazana na rys. 2 ma zdaniem autora, charakter uniwersalny.



Rys. 2. Struktura potrzeb w zakresie realizacji prac typu projektowego (D1), weryfikacyjnego (D2) i uzupełniającego (D3)

Zaspokojenie tak określonych potrzeb w sposób optymalny nie jest oczywiście możliwe jednym urządzeniem, lub jednym poziomem automatyzacji — wchodzą tu w grę poziomy tworzące układy automatyzacji prac, które można scharakteryzować następująco:

A. Struktura potrzeb dla prac typu projektowego obejmuje wszystkie stopnie złożoności obliczeń, co wymaga zaangażowania wszystkich poziomów automatyzacji. Powstały układ powinien zapewniać czas oczekiwania na wyniki nie przekraczający 2—3 godzin, oraz możliwość wejścia na maszynę, praktycznie o każdej porze dnia, bowiem możliwość planowania obliczeń typu projektowego (a więc i realizacji obliczeń w wyznaczonym czasie) jest niewielka. Wymaga to:

— na poziomie podstawowym: dostatecznej ilości i właściwego rozmieszczenia kalukulatorów programowanych, obsługujących określone zespoły użytkowników; według Millera i McCastlina, na jedno takie urządzenie powinno przypadać 3 użytkowników (projektantów)

— na poziomie średnim: łatwego dostępu do końcówki maszyny wieloprogramowej lub bezpośredniego dostępu do małego komputera o wielkości rzędu 64—128 Kb, przy czym uważa się, że praca w systemie wsadowym jest niepożądana dla tego typu obliczeń zarówno ze względu na trudności w zapewnieniu krótkiego czasu oczekiwania na wyniki jak i ze względu na brak kontroli użytkownika nad realizacją obliczenia (zamknięty system pracy) [2]

— na poziomie wyższym: dostęp do końcówki maszyny wieloprocesowej, przy czym uważa się, że dolną granicą pojemności pamięci takiej maszyny, umożliwiającą pracę w trybie „on line" w zakresie odpowiednio zorganizowanych obliczeń projektowych, jest 500 Kbajtów [2].

W zakresie formy wyników należy zwrócić szczególną uwagę na zapewnienie wyjścia graficznego, szczególnie przydatnego w pracach projektowych i to na wszystkich poziomach automatyzacji.

B. Prace typu weryfikacyjnego, obejmujące dwa górne stopnie złożoności obliczeń nie wykorzystują w zasadzie podstawowego poziomu automatyzacji. Wymagany układ powinien zapewnić czas oczekiwania na wyniki w granicach jednej doby (jednego dnia pracy), w związku z czym istnieje możliwość planowania pracy w skali dnia. Różnorodność obliczeń jest dość duża przy preferencji średniego poziomu automatyzacji, przy czym konieczny jest także dostęp do dużego systemu, z pamięcią o pojemności powyżej 250 Kbajtów, ale może on być realizowany w systemie wsadowym. Potrzeby tego typu spełnia w zasadzie maszyna wieloprogramowa z pamięcią do 500 Kbajtów użytkowana z końcówki lub wsadowo w zależności od potrzeby.

C. Prace typu uzupełniającego zawierają się w dwóch dolnych stopniach złożoności obliczeń i nie wykorzystują z ko-

lei wyższego poziomu automatyzacji. Wymagany czas oczekiwania na wyniki zawiera się w granicach kilku dni, przy dużej różnorodności obliczeń i form wyników. Prace tego typu mogą być realizowane w określonych przedziałach czasowych i w systemie zamkniętym.

Naszkicowana powyżej charakterystyka układów automatyzacji jest punktem wyjścia do określenia układu generalnego, obejmującego pełną strukturę prac (wszystkie rodzaje działań), właściwą dla dużej organizacji projektowej. W takim układzie obejmującym wszystkie poziomy automatyzacji i ich cechy wyszczególnione w punktach A, B i C optymalna konfiguracja urządzeń powinna zawierać:

— znaczną liczbę kalkulatorów programowanych realizujących prace typu projektowego i uzupełniającego w poważnym zakresie pracochłonności (do 50% wszystkich prac)

— maszynę wieloprogramową z pamięcią do 250 Kbajtów realizującą prace wszystkich typów zarówno w zdalnym dostępie jak i bezpośrednio przy zróżnicowanych czasach oczekiwania na wyniki; można przewidywać znaczne obciążenie takiej maszyny ze względu na jej wielofunkcyjność, pod warunkiem, że liczba końcówek zdalnego dostępu nie będzie zbyt duża, co może wydłużyć znacznie czas oczekiwania na wyniki a także — zmniejszyć wielkość dostępnej pamięci

— pewną liczbę końcówek do dużego systemu wieloprocesowego z pamięcią powyżej 500 Kbajtów realizującego przede wszystkim prace typu projektowego o dużym stopniu złożoności w trybie „on line" przy minimalnych czasach oczekiwania na wyniki.

Duży system wieloprocesowy mógłby być użytkowany przez kilka organizacji projektowych.

Projektowanie automatyzacji

W świetle przedstawionego podejścia do zadania automatyzacji prac projektowych, zaprojektowanie takiej automatyzacji wymaga określonych działań rozpoznawczych o niewątpliwej randze badawczej, zwłaszcza, jeżeli projektowany układ ma mieć szerszy zasięg, np. o skali branżowej, obejmując szereg organizacji (środowisk) projektowych. Kierunki badań dotyczą trzech wzajemnie uwarunkowanych dziedzin:

a) dziedziny rozpoznania i ustalenia struktury środowisk użytkowników w celu doboru możliwie najlepszego układu generalnego automatyzacji i jego organizacji, co jest warunkiem uznania przez użytkownika proponowanego mu układu za przydatny do jego celów i odpowiedzią na pytanie, GDZIE automatyzować

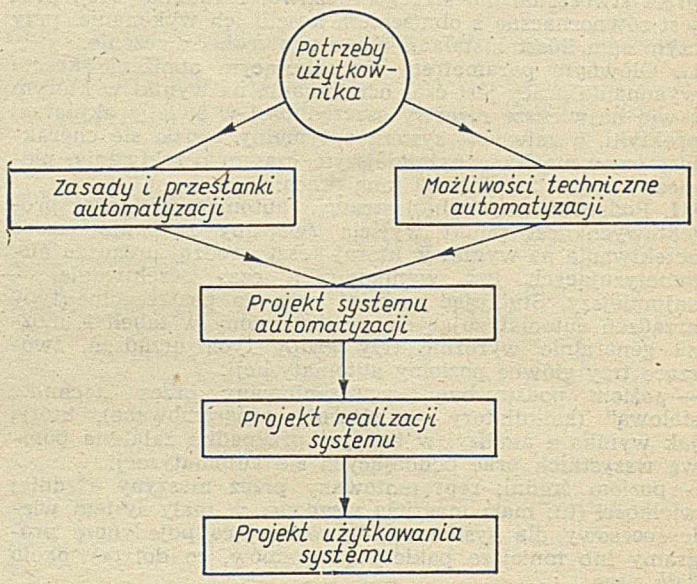
b) dziedziny identyfikacji prac projektowych, ich zakresu, złożoności i przebiegu w celu doboru właściwych narzędzi automatyzacji, co jest warunkiem skuteczności działania proponowanych użytkownikowi rozwiązań i odpowiedzią na pytanie, CO automatyzować

c) dziedziny usprawniania komunikacji na drodze człowiek-maszyna w celu maksymalnej eliminacji z pola widzenia użytkownika problemów technicznych związanych z realizacją obliczeń, co jest warunkiem stosowania uznanych i skutecznych rozwiązań w tych sytuacjach dla których zostały one przewidziane i odpowiedzią na pytanie, JAK automatyzować.

Dwie pierwsze dziedziny przygotowują organizacyjne i techniczne warunki działania układów automatyzacji prac — o czym była już mowa w poprzednich rozdziałach. Warto teraz przyjrzeć się nieco trzeciej dziedzinie, która w największym stopniu decyduje o sprawności układu, organizując współpracę człowiek-maszyna.

Postawiony na wstępie postulat projektowania systemów[1]) automatyzacji ukierunkowany na walory użytkowe tych systemów — mimo pozornego truizmu takiego sformułowania — pociąga określone konsekwencje praktyczne. Pierwszym zadaniem jest tu opracowanie możliwie pełnego projektu takiego systemu na co składają się trzy główne elementy: sam system, posób jego realizacji i sposób jego użytkowania — rys. 3

Wydzielenie tych właśnie a nie innych elementów wynika z ich decydującej roli w uzyskaniu użytkowych walorów rozwiązania. Projekt samego systemu — w przyjętej ogólnie praktyce — zawiera poza uporządkowaną, zalgorytmizowaną procedurą postępowania przekładaną na język programu, formalne przepisy komunikacji użytkownika z maszyną (forma i zakres danych i wyników) oraz przepisy



Rys. 3. Elementy projektu automatyzacji prac projektowych

eksploatacji technicznej systemu, warunkujące jego działanie na określonym typie maszyny. Są to z pewnością elementy konieczne dla automatycznej realizacji prac objętych programem, ale czy wystarczające?

Przyjęty sposób realizacji systemu w dużym stopniu decyduje o tym, czy będzie on wykorzystywany i aktualny. Pierwsza wątpliwość wynika z faktu, że — jak pisze Rajchert — systemy rozwojowe tworzone są głównie przez specjalistów związanych z ośrodkiem obliczeniowym, co powoduje, że automatyzacja prac prowadzona jest „z góry" jako działalność oderwana od środowiska projektowania, przestaje natomiast być procesem rozwijającym się organicznie, oddolnie. Prowadzi to do powstawania klanu specjalistów, którym musi podporządkować się projektant a jego pozycja zostaje zdominowana przez technikę obliczeniową [2].

Taki sposób realizacji systemu jest w sensie społecznym z góry skazany na niepowodzenie, gdyż projektanci po prostu nie uznają narzuconych im z góry rozwiązań. Druga wątpliwość wypływa z dość powszechnej praktyki projektowania dużych systemów bez dostatecznego zabezpieczenia wykonawczego i niezbędnego doświadczenia realizacyjnego, co powoduje niekorzystne wydłużanie się terminów oddania systemu do użytku. Bywają przypadki, że takie systemy całymi latami „są w trakcie realizacji" przy czym, oczywiście, po pierwszych kilku latach przestają być aktualne. Propozycja właściwego i raczej decydującego udziału użytkownika w projektowaniu systemu i sposobu jego wykorzystywania oraz ocena możliwości wykonania pracy w rozsądnym czasie, powinny być głównym akcentem tego etapu projektu systemu.

Sposób realizacji zwany potocznie „od małej mechanizacji do dużej automatyzacji" wydaje się metodologicznie najbardziej właściwy.

Jest to kierunek naturalnego rozwoju systemów obliczeniowych od stosunkowo prostych prac podstawowego poziomu automatyzacji, łatwych do realizacji i nie wymagających złożonej organizacji do prac trudniejszych, wymagających większego doświadczenia i wysiłku realizacyjnego.

Jest to automatyzacja „w kierunku nadmiaru" — na wyższy poziom automatyzacji przechodzą te prace, które w sposób oczywisty nie mieszczą się na poziomie niższym. Dzięki temu dokonuje się także naturalna selekcja prac z których tylko część wymaga wyższego poziomu automatyzacji a cały proces realizacji przebiaga stosunkowo łagodnie, stopniowo, szerokim frontem działania. Kierunek

[1]) przyjęte w dalszym tekście określenie „system" oznacza odpowiednio zaprogramowany proces automatyzacji pracy w postaci programu lub układu programów.

Tabela III. Przykład oceny użytkowej systemu

A. **Ogólna charakterystyka systemu;** Program na obliczanie sił wewnętrznych w kadłubie okrętu od obciążeń wzdłużnych na spokojnej wodzie, opracowany na komputer ICL 4/70 z pamięcią wewnętrzną 256 Kbajtów i posiadający skomponowane na stałe 12 wariantów prowadzenia obliczeń. Program jest eksploatowany wsadowo (karty); wymagana pojemność pamięci wynosi 137 Kbajtów — wyjście na drukarkę z możliwością archiwowania danych i wyników na taśmie magnetycznej.

B. **Ocena użytkowa systemu**

Nr kryterium	Komentarz do oceny	Ocena w trójstopniowej skali		
		0	1	2
1	program był projektowany na maszynę średniej wielkości — „niedomiar" nie występuje			
2	układ danych jest dość sformalizowany i mógłby być lepiej rozwiązany			
3	wyniki są przejrzyście zaprojektowane, ale brak pożądanego bardzo wyjścia graficznego z pełnym opisem			
4	forma użytkowania (wsadowa) nie zawsze zapewnia otrzymanie wyników na drugi dzień, co jest wymagane przy tym programie			
5	nie występują „zacięcia" w realizacji samego programu			

C. **Wnioski.** W celu poprawy cech użytkowych programu należy:
a) udoskonalić organizację danych wejściowych, przechodząc na bardziej swobodny zapis informacji
b) zaprojektować wyjście graficzne (wykresy sił wewnętrznych) z opisem arkuszy
c) dostosować program do pracy w zdalnym dostępie.

Tabela II. Kryteria oceny cech użytkowych systemu

Lp.	Kryterium	Nadana wartość oceny w trójstopniowej skali		
		0	1	2
1	**Sprawność instrumentalizacyjna**			
	— wielkość i struktura systemu zgodna z charakterem narzędzia automatyzacji;			x
	— sytuacja „niedomiaru": realizacja małych problemów na dużych maszynach;		x	
	— sytuacja „nadmiaru": sztuczny podział dużych algorytmów celem realizacji na małych urządzeniach.	x		
2	**Sprawność wejścia**			
	— przejrzysta, elastyczna organizacja wprowadzania danych;			x
	— układ danych o znacznym stopniu formalizacji, wymagający znacznej pracy przygotowawczej;		x	
	— j.w., oraz dodatkowo wymagający wstępnych przeliczeń pomocniczych.	x		
3	**Sprawność wyjścia**			
	— wyniki mają charakter ostateczny i nie wymagają dodatkowej obróbki;			x
	— wyniki wymagają opracowania innej formy zapisu (np. graficznej) i dodatkowego opisu;		x	
4	**Sprawność odpowiedzi**			
	— czas odpowiedzi zgodny z wymaganiami procesu projektowania;			x
	— otrzymanie wyników na czas wymaga działań interwencyjnych;		x	
	— otrzymanie wyników na czas nie jest praktycznie możliwe.	x		
5	**Sprawność operacyjna**			
	— system działa niezawodnie przy każdej przewidzianej konfiguracji danych i wyników;			x
	— system ma „słabe punkty" w pewnych sytuacjach eksploatacyjnych		x	
	— system wymaga z reguły wielokrotnych wejść dla jednej realizacji zadania.	x		

przeciwny — także spotykany w praktyce — nie posiada tych walorów.

Użytkowanie systemu powinno być szczególnym obiektem projektowych rozważań, od tego bowiem w jakim stopniu proponowane rozwiązanie automatyzacji określonej pracy będzie zmuszało użytkownika do wykonywania dodatkowych czynności związanych z formalizacją problemu lub do oczekiwania na wyniki, zależy wykorzystanie i efektywność tej automatyzacji. Wydaje się, że w większości przypadków sposób użytkowania jest raczej wynikiem splotu różnych okoliczności — głównie technicznej natury — wynikłych w trakcie realizacji lub eksploatacji programu, niż świadomym, przemyślanym elementem projektu systemu. Projektując użytkowanie systemu należy zatem zwrócić szczególną uwagę na te czynniki, które w największym stopniu wpływają na proces pracy użytkownika-projektanta. Można tu wymienić:

— stopień zgodności struktury i wielkości systemu z charakterem narzędzia automatyzacji, co nazwano sprawnością instrumentalizacyjną

— zakres swobody w formułowaniu zadania (wprowadzaniu danych), co nazwano sprawnością wejścia

— stopień gotowości wyników (w sensie uzyskania pełnej, końcowej odpowiedzi), co nazwano sprawnością wyjścia

— czas oczekiwania na wyniki w stosunku do wymagań procesu projektowania, co nazwano sprawnością odpowiedzi

— niezawodność działania systemu w normalnych warunkach eksploatacji, co nazwano sprawnością operacyjną.

Ocena poszczególnych czynników, które można nazwać cechami użytkowymi systemów z punktu widzenia użytkownika-projektanta, pozwala na kontrolę stopnia realizacji trzeciego, ostatniego elementu projektu automatyzacji prac, pokazanego na rys. 3. W celu lepszego zilustrowania cech użytkowych systemu pożyteczne może być wprowadzenie pseudoilościowej skali ocen, np. w postaci przedstawionej w tabeli II. Daje to możliwość graficznego przedstawienia takiej charakterystyki, jak to pokazano w tabeli III na przykładzie programu średniej wielkości ocenianego jako samodzielna jednostka o charakterze weryfikacyjnym, do której to funkcji został zaprojektowany. Ocena taka — mimo subiektywizmu właściwego wszystkim ocenom pseudoilościowym — zmusza projektanta systemu do krytycznego spojrzenia na użytkowe cechy koncepcji rozwiązania, co pozwala na uniknięcie poważnych wad w tym zakresie, zanim użytkownik natknie się na nie w trakcie eksploatacji systemu, kiedy jest z reguły za późno na zasadnicze zmiany. Można tu także przyjąć, że uzyskanie „zerowej" oceny dla któregokolwiek kryterium dyskwalifikuje system pod względem użytkowym.

Zakończenie

Poruszona w artykule problematyka, związana ze strategią automatyzacji prac projektowych jest odbiciem praktycznych trudności na jakie napotykają inżynierowie związani z oprogramowaniem prac projektowych. Byłoby ze wszech miar pożądane, aby problemy te dały się uporządkować i przenieść ze sfery naprawy rzeczywistości, która często zaskakuje odmiennością od początkowych wyobrażeń o tym co się da zrobić w informatyce, traktowanej jako uniwersalny środek na wszystkie bolączki projektowania, do sfery działań projektowych, uprzedzających niepożądane fakty i przewidujących właściwe rozwiązania w dobrze rozpoznanym obszarze planowanej zmiany procesów informacyjnych.

Jako generalny wniosek można więc przedstawić oczywistą chyba konkluzję, że automatyzacja prac projektowych też musi zostać zaprojektowana ze wszystkimi konsekwencjami tego procesu w którm użytkownik odgrywa decydującą rolę, a także ponosi najwięcej niepowodzeń, jeżeli rozpoznanie sfery jego działania nie było dostateczne.

Literatura:

[1] Ralph E. Miller, John McCaslin: Cost-effective matching of task and computing device. Journal of the Structural Division ASCE, vol. 97, No STi, Jan. 1971.

[2] Paul Rajchert: Problemy zastosowania urządzeń elektronicznej techniki obliczeniowej do projektowania w budownictwie. Stała Grupa Robocza d/s zastosowań metod matematycznych i techniki obliczeniowej w budownictwie w Stałej Komisji Budownictwa RWPG, Sopot, 1973.

ALEKSANDER KUŹMIN
Zakład Elektronicznej Techniki Obliczeniowej
Gdynia

681.322.06:658.51

Normatywy programowania

Przedstawiono metodę określania czasu normatywnego dla realizacji prac programowych, która została ustalona na podstawie publikacji zagranicznych oraz doświadczeń praktycznych ZETO Gdynia.

Przedstawiona w pracy tej metoda ustalenia czasu normatywnego dla realizacji prac programowych, opracowana została na podstawie publikacji VVB (Vereinigung Volkseigeur Retriebe) Maschinelles Rechnen będącego na terenie NRD odpowiednikiem naszych ZETO.

Wskaźniki i normatywy zawarte w niniejszym opracowaniu w części zostały zaczerpnięte z wyżej wspomnianych publikacji, w części natomiast zostały przystosowane do naszych warunków.

Wskaźniki zawarte w niniejszym opracowaniu odpowiadają średniemu poziomowi wiedzy i doświadczeniu kadry informatyków zatrudnionych w ośrodku obliczeniowym. Oczywistym jest, że normatywy przytoczone w tej pracy nie stanowią ostatecznych rozwiązań w tej kwestii i ulegną zmianie z chwilą podniesienia się u pracowników poziomu wiedzy informatycznej i doświadczenia.

Zakres skuteczności metody

Z doświadczenia wynika, że z problemem szacowania czasu realizacji prac programowych spotykamy się trzykrotnie (w 3 etapach) w czasie trwania pracy nad oprogramowaniem systemu.

Etap pierwszy — oszacowanie czasu realizacji prac programowych w stadium projektowania założeń systemu. Oszacowanie na tym etapie jest szacunkiem przybliżonym.

Etap drugi — to etap prac, na którym dysponujemy już kompletnymi założeniami do programów. Oszacowanie na tym etapie jest bardziej dokładne i uwzględnia normy odpowiadające średniemu poziomowi wiedzy i doświadczenia istniejącego zespołu programistów.

Etap trzeci — to etap prac w którym powierzamy określony program do napisania konkretnemu programiście. Szacując czas realizacji prac programowych na tym etapie uwzględniamy indywidualne kwalifikacje programisty i jego specjalistyczną znajomość problemu.

Określone tu etapy różnią się między sobą różnym stopniem dokładności oszacowania.

Przy ustaleniu czasowych wskaźników normatywnych należy wziąć pod uwagę przeciętny czas realizacji prac programowych.

Opis metody

Wychodząc z założenia, że programowanie jest pracą umysłową twórczą i jako takie trudne jest w ocenie jakościowej w niniejszej metodzie przyjęto kryterium ilościowe.

Wyróżniono nadto następujące fazy **realizacji prac programowych** (RPP):

— wprowadzenie do określonego problemu
— studiowanie założeń programowych i omówienie pewnych niejasności — wynikających z niedoskonałości założeń — z programistą wiodącym
— układanie schematu blokowego programu i jego sprawdzenie
— kodowanie programu i sprawdzenie kodowania
— przygotowanie danych testujących do programu
— kompilacja i testowanie programu
— sporządzenie dokumentacji programowej.

Opisywana metoda uwzględnia przy opracowywaniu programu następujące czynniki:

a) obiektywne warunki rozwiązania problemu (stopień trudności i ujęcia programu), wyrażające się w wartościach wskaźnika E i I

b) subiektywne warunki, takie które wywierają trwały wpływ na opracowanie programu

a więc stosunek aktualnej i wymaganej wiedzy informatycznej do rozwiązania problemu (wyrażający się we wskaźniku W) i poziom kwalifikacji danego programisty (wyrażający się we wskaźniku Q).

Opis etapów szacowania czasu RPP

Ustalenia wynikłe z oszacowania czasu RPP na etapie II, nie powinny ulec zmianie w etapie III, gdyż związana z tym zmiana kosztów nie miałaby uzasadnienia.

Oczywistym jest, że dopuszczalne są zmiany w oszacowaniu czasu RPP na poszczególnych etapach lecz tylko w niewielkim stopniu.

Oszacowanie czasu RPP na etapie projekotwania. Etap I. Oszacowanie czasu RPP dla określonych programów służy na tym etapie do ustalenia terminu zakończenia prac ze zleceniodawcą i do opracowania harmonogramu realizacji tych prac.

Czas RPP na tym etapie szacujemy przy pomocy wzoru:

$$K = (E + I) \cdot W \cdot T$$

Objaśnienia:

K — całkowity czas RPP w godzinach

E — ilość środków wej/wy w programie

(np. wejście w programie — 2 MT i 1 CR = 3, wyjście — 1 MT, 2 LP, 1 TP = 4) [1] wówczas wskaźnik $E = 3 + 4 = 7$

Uwaga

Przy równoczesnym przetwarzaniu różnych rodzajów kart, każdy rodzaj karty będzie oznaczał niezależny środek wejścia, dotyczy to również drukarki wierszowej z różnymi formatami wydruku itp.

I — jest współczynnikiem trudności programu wyrażonym w tabeli I. Na tym etapie współczynnik I będzie można określić tylko w przybliżeniu (tab. I).

Tabela I. Stopień trudności programu

Wskaźnik	Prosty	Średni	Trudny
I	do 5	do 11	do 27

[1]) stosowane w nawiasie skróty oznaczają:
MT — jednostka taśmy magnetycznej
CR — czytnik kart
LP — drukarka wierszowa
ITP — dziurkarka taśmy

Wskazane jest by przy określaniu przybliżonego stopnia trudności programu wybierać wartości pośrednie.
Górna granica 27 jest praktycznie nie stosowana.
W — charakteryzuje stan wymaganej i aktualnej wiedzy informatycznej (specjalistycznej) potrzebnej do oprogramowania danego programu. Podobnie jak wskaźnika *I* i tu w stadium projektowania możliwe jest tylko przybliżone określenie wartości wskaźnika *W*.

Tabela II. Wymagana wiedza informatyczna

Wskaźnik	Niewielka	Średnia	Duża
W	2,50	2,75	3,00

Tabela III

Elementy programowania	Stopień trudności		
	Prosty	Średni	Skomplikowany
1. Operacje arytmetyczne	1	3	5
2. Przesunięcia i zapamiętywania	1	3	4
3. Kontrola błędów, sprawdzania	1	4	7
4. Czytanie tabel, wyszukiwania	2	5	8

Wartości wskaźnika *W* na tym etapie są określone w sposób podany w tabeli II.

Przy czym będziemy tu rozumieli przez WYMAGANA WIEDZA INFORMATYCZNA nie tylko znajomość programowania lecz również wymaganą znajomość problemu, którą programista musi wypracować i przyswoić.
T — jest wartością stałą, wskazuje ona liczbę możliwych godzin pracy programisty.

Wskaźnik $T = 7{,}6$
wynika to z następujących obliczeń.
Liczba godzin pracy w tygodniu 46
46 : 6 liczbę dni w tyg. = 7,6

Oszacowanie czasu RPP po opracowaniu szczegółowych założeń programowych. Etap II. Na etapie możliwe jest dokładne oszacowanie czasu RPP, gdyż dysponujemy już dokładną znajomością stopnia trudności problemu oraz szczegółowymi założeniami programu. Określając wskaźniki *W* i *I* musimy pamiętać o uwzględnieniu przeciętnej wiedzy informatycznej pracownika.

Na etapie tym opracowujemy również harmonogram wykonywania prac programowych, który będzie podstawą do określenia kosztów tych prac. Szacowanie czasu RPP odbywa się przy zastosowaniu norm bez względu na to, który programista będzie opracowywać program.

Czas RPP szacujemy wg wzoru podanego powyżej (Etap I) przy czym jak już wspomniano, możliwe jest dokładniejsze określenie wskaźników *I* i *W*. W tabeli III zawarte są szczegółowe wskaźniki I dla poszczególnych elementów programowania.

Wskaźnik *I* równa się sumie poszczególnych elementów programowania. Tabela IV obrazuje wskaźnik *W:* jest to stosunek wymaganej do aktualnie posiadanej znajomości problemu przez zespół programowy.

Tabela IV

Aktualna wiedza informatyczna	Wymagana wiedza informatyczna		
	mała	średnia	duża
Duża praktyka w programowaniu oraz bardzo dobra znajomość problemu	2,00	2,25	2,75
Duża praktyka w programowaniu oraz poprawna znajomość problemu	2,00	2,50	3,00
Duża praktyka w programowaniu oraz brak znajomości problemu	2,00	2,50	3,25
Pewna praktyka w programowaniu oraz poprawna znajomość problemu	2,25	3,00	3,75
Pewna praktyka w programowaniu oraz brak znajomości problemu	2,50	3,25	4,00

Oszacowanie czasu RPP przy uwzględnieniu indywidualnych cech programisty. Etap III. Szacując czas RPP uwzględniamy cechy indywidualne programisty, a więc jego staż i doświadczenie. Wzór na oszacowanie czasu RPP w tym stadium ma nieco inną postać i tak:

$$K = (E + I) \times W \times T \times Q$$

gdzie *Q* jest współczynnikiem kwalifikacji programisty. Znaczenie współczynników *E*, *I* oraz *W* opisane jest przy omawianiu szacowania RPP w etapie I.

Tabela V

Stanowisko	Współczynnik kwalifikacji
St. programista analityk	0,7
St. programista	0,8
programista	1,0
Mł. programista	1,2
stażysta (asystent)	1,4

Uwagi końcowe

Problemem oszacowania czasu RPP zajmować powinni się tylko doświadczeni pracownicy (główny analityk, st. programista analityk).

Wszystkie współczynniki powinny odpowiadać średniemu poziomowi wiedzy informatycznej.

Opracowanie to nie rości sobie pretensji do definitywnego rozstrzygnięcia problemu, jakim jest szacowanie czasu realizacji prac programowych, a jest jedynie sygnałem o coraz częściej występującym zapotrzebowaniu na metodę ustalenia czasu normatywnego dla prac programowych.

Próby ustalenia takich metod podjęto w ZETO Gdynia. Autor niniejszego opracowania wdzięczny byłby Czytelnikom za ewentualne uwagi krytyczne i dyskusyjne.

KRZYSZTOF GAJOS
Warszawa

SESTEL system ewidencji środków telekomunikacji

W artykule omówiono modelowy wielodostępny system ewidencji środków technicznych sieci telekomunikacyjnych opracowany na maszynę SINGER-10.

Sieć telekomunikacyjna jest organizmem, który wymaga ciągłego nadzoru technicznego. Od sprawności funkcjonowania tego organizmu w znacznej mierze zależy sprawne funkcjonowanie gospodarki kraju.
Decydujący wpływ na pracę sieci ma stan techniczny jej urządzeń. Uszkodzenia pociągają za sobą pewne zmiany w jej konfiguracji. Wszelkie bieżące modyfikacje układu sieci muszą być ewidencjonowane w odpowiednich dokumentach. W wyniku tego, określenie stanu sieci, jej rezerw i możliwości w danej chwili czasowej jest bardzo utrudnione.

Założenia systemu

W Pracowni Teleinformatyki Centralnego Laboratorium GUTM opracowany został modelowy system komputerowy dla potrzeb ewidencji środków technicznych sieci telekomunikacyjnej o nazwie „SESTEL". System wykorzystuje komputer SINGER-10. Jest to maszyna pracująca z podziałem czasu. Umożliwia ona jednoczesną pracę 20 użytkowników, z których każdy ma do dyspozycji 1 sekcję pamięci operacyjnej o pojemności do 10 K znaków oraz określony obszar pamięci wspólnej. Do każdej sekcji można dołączyć wszystkie podstawowe urządzenia zewnętrzne, jak drukarka wierszowa, czytnik kart itp., które dostępne są tylko dla danej sekcji. Komunikacja między sekcjami odbywać się może albo poprzez pamięć wspólną, albo poprzez pamięć dyskową czy taśmową, z których korzystają wszystkie sekcje. Procesor obsługuje sekcje pamięci kolejno pod nadzorem układowego systemu operacyjnego. Czas pracy jednej sekcji pamięci wynosi średnio 37 ms.
Przyjęto założenie, że SESTEL będzie systemem wielodostępnym i każdy z abonentów będzie mógł wykonać na zbiorach pewne ściśle określone operacje. Jednocześnie system powinien umożliwiać pracę konwersacyjną.
Każde stanowisko powinno być wyposażone w jedno urządzenie konwersacyjne, w tym przypadku najlepiej w monitor ekranowy.
Ponadto stanowiska, które wymagają tworzenia trwałej dokumentacji wyposażone będą w drukarkę znakową.

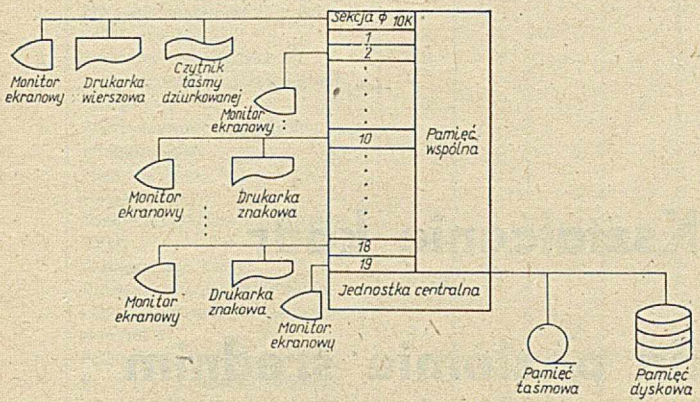
Mgr inż. KRZYSZTOF GAJOS ukończył Wydział Elektroniki Politechniki Warszawskiej (1973 r.) o specjalności maszyny matematyczne. Pracował w Ośrodku Badawczo-Rozwojowym Techniki Medycznej ORMED. Obecnie zatrudniony jest na stanowisku specjalisty w Pracowni Teleinformatyki Centralnego Laboratorium Głównego Urzędu Telekomunikacji Międzymiastowej.

Urządzenia te powinny być rozmieszczone w różnych komórkach organizacyjnych Urzędu, tzn. tam, gdzie informacja o stanie elementów sieci telekomunikacyjnej ma swoje źródło oraz tam, gdzie dostęp do tej informacji jest konieczny (np. dla celów zarządzania).
W celu zapewnienia właściwej ochrony banku informacji, system ma możliwość identyfikacji urządzeń z nim współpracujących. Umożliwia to ochronę przed niepowołanym dostępem.
Poszczególne urządzenia zewnętrzne uprawnione będą do wykonywania z nich tylko określonych operacji (np. z określonych urządzeń końcowych nie można modyfikować informacji zawartych w banku systemu).
Każdorazowo, pod koniec dnia pracy, zawartość banku informacji powinna być skopiowana na taśmę magnetyczną. Dzięki temu możliwe jest odtworzenie zbioru danych w przypadku ich uszkodzenia czy zniszczenia. Również wszelkie manipulacje zawartością zbioru powinny być rejestrowane na taśmie magnetycznej i, na żądanie operatora, drukowane. Istnieje więc możliwość ustalenia kto, kiedy i jakich operacji dokonywał na zbiorze.

Opis systemu SESTEL

System SESTEL operuje na następujących zbiorach danych:
— zbiór łączy
— zbiór grup pierwotnych
— zbiór grup wtórnych
— zbiór grup trójnych
— zbiór kodów i nazw miast (zgodnie z kodami TELKO).
Łącze jest to pasmo częstotliwości zawarte w granicach pasma akustycznego (300 Hz — 340 Hz). Aby po jednym torze przesłać maksymalną liczbę kanałów, łącza te poddaje się modulacji, przenosząc je w ten sposób w inny zakres częstotliwości.
Grupą pierwotną nazywa się grupę łączy po pierwszej takiej modulacji. Modulując dalej, otrzymuje się z 5 grup pierwotnych grupę wtórną, a następnie podobnie grupę trójną.
Wszystkie zbiory, o których wspomniano wyżej, są zgromadzone na dyskach. W obrębie każdego zbioru operuje się adresami względnymi poszczególnych zapisów (rekordów).
Dzięki temu osiągnąć można krótszy czas dostępu, co jest istotne przy manipulacji tak dużą liczbą danych.
Korzystając z powyższych danych, SESTEL zapewnia uzyskanie następujących informacji:
— budowa i przebieg trasowy łączy oraz ich parametry
— wyszczególnienie łączy wg określonej relacji (również z podziałem wg typu)
— zajętość kanałów grupy pierwotnej
— zajętość kanałów grupy wtórnej
— zajętość kanałów grupy trójnej
— położenie grupy modulacyjnej w grupie modulacyjnej wyższej krotności
— zajętość kabla określonego typu (wykaz grup modulacyjnych pracujących w danym kablu)
— schemat blokowy przebiegu łącza trasowego.
Źródło informacji dla powyższych zbiorów stanowi karta ewidencyjna łącza oraz karta zajętości urządzeń. Informacje zawarte w pierwszym z tych dokumentów są wystarczające z punktu widzenia EPD, drugi dokument natomiast należy w przyszłości zmienić pod kątem potrzeb EPD.

Cała informacja niezbędna do utworzenia zbiorów, a zawarta w dokumentach źródłowych, nanoszona jest na taśmę 5-ścieżkową i stanowi podstawowy nośnik informacji. Jest to ważny szczegół, gdyż dalekopis jest bardzo powszechnym i łatwo dostępnym urządzeniem w GUTM-ie. Taśma ta po odczytaniu i sformatyzowaniu zawartych w niej danych w zapisy (rekordy) o stałej długości 80 znaków zostaje zapisana na dysku, gdzie każdy rekord zajmuje 1 sektor.



Rys. 1. Konfiguracja komputerowa SINGER-10 dla systemu SESTEL

Konwersacja z systemem

Użytkownik komunikuje się z systemem za pomocą klawiatury monitora ekranowego. Język systemu jest prosty, umożliwiający szybką i prostą obsługę, nawet przez ludzi bez praktyki w zakresie współpracy z komputerem. System zapewni również dużą swobodę syntaktyczną przy budowaniu zlecenia.

System akceptuje następujące zlecenia:
W — wyświetlenie opisu łącza
M — modyfikacja opisu łącza
R — wyświetl i wydrukuj opis łącza należącego do określonej relacji
Z — wyświetl i wydrukuj zajętość grup niższego rzędu w grupach wyższego rzędu
ZK — wyświetl i wydrukuj zajętość kabla (grupy pierwotne pracujące w danym kablu)
B — narysuj na ekranie monitora schemat blokowy przebiegu łącza.

Przykładowa struktura zlecenia przedstawia się następująco:

OP [operand 1 — operand 2] operand 3

gdzie:

OP — kod operacji
operand 1, 2 — nazwa relacji, np. Warszawa—Katowice
operand 3 — typ łącza/grupy.

Operandy 1, 2 określają nazwy miejscowości, gdzie odpowiednio rozpoczyna się i kończy relacja poszukiwanej grupy. W przypadku zlecenia z kodem ZK, operand 1 stanowi literową część numeru eksploatacyjnego kabla.

Operand 3 to numer eksploatacyjny grupy lub łącza.

Dla operacji z kodem Z podaje tu się typ szukanych łączy (automatyczne, półautomatyczne, ręczne).

W przypadku zlecenia z kodem W, na ekranie monitora wyświetlają się dane techniczne wybranego łącza wraz z komentarzem słownym, opisującym znaczenie poszczególnych jego parametrów.

Modyfikując informacje o łączu, operator używa rozkazu z kodem M. W odpowiedzi wyświetla się informacja z opisem, jak w zleceniu W. Modyfikując rekord łącza, operator podaje ewentualnie tylko część zmian, a reszta informacji, która powiązana jest z nowowprowadzonymi poprawkami zostanie automatycznie zmieniona. Po wprowadzeniu zmian, wyświetlona jest ponownie cała informacja o łączu celem ewentualnego skontrolowania poprawności operacji.

W przypadku zlecenia B, na ekranie rysowany jest schemat blokowy przebiegu łącza, dając tym samym przekrojową informację o nim.

Podsumowanie

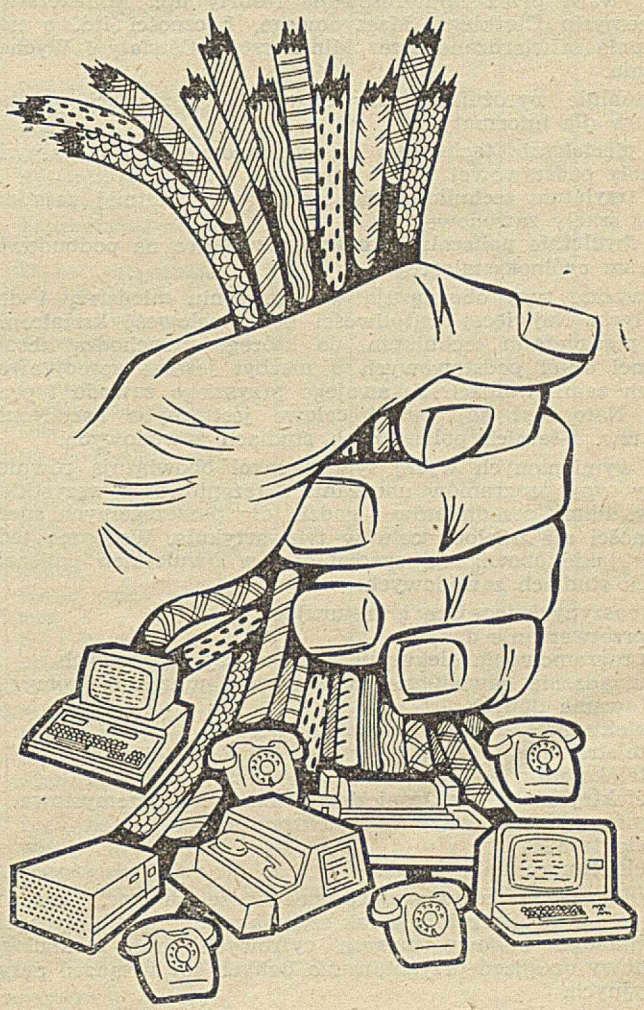
Jak wspomniano na wstępie, SESTEL jest systemem modelowym. Model opracowany został na maszynę SINGER-10 i zajmuje sekcję o pojemności 10 K znaków oraz część pamięci wspólnej.

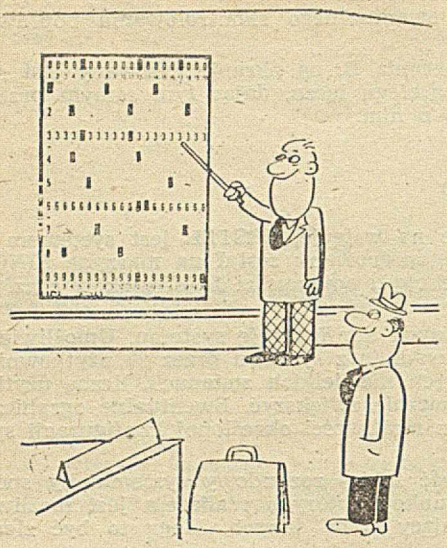
Rysunek przedstawia konfigurację systemu. Umożliwia ona jednoczesne korzystanie z systemu przez 20 użytkowników, choć przy pewnych niewielkich zmianach, liczbę użytkowników można znacznie zwiększyć. Ewentualne ograniczenia wynikają tylko z możliwości określonej konfiguracji sprzętu.

Istnieje możliwość takiej rozbudowy systemu, by spełniał on dodatkowe funkcje, których realizacja jest możliwa w oparciu o istniejący zbiór danych. Mogą to być zlecenia dotyczące:

— stanu liczbowego łączy (z podziałem na rodzaje)
— określenia liczby łączy uruchomionych w ostatnim okresie
— określenia wolnych kanałów w danej grupie modulacyjnej
— liczby i jakości grup modulacyjnych w danej relacji
— określenia zajętości danego systemu telekomunikacyjnego
— rysowania przebiegu trasowego w grupie modulacyjnej.

Rozbudowa systemu jest możliwa ze względu na jego modularny charakter. Istniejące w tym sensie rezerwy możliwości systemu SESTEL wykorzystane zostaną przy uruchomieniu pełnego systemu użytkowego.





Kształcenie kadr na poziomie średnim

BOGUMIŁ STACHURA
Ministerstwo Oświaty i Wychowania

Obecne kształcenie średniej kadry technicznej informatyków jest oparte o programy nauczania, które były opracowane w latach 1965—72.

Wiele z tych programów prezentuje dość zróżnicowane treści i formy kształcenia. Programy były najczęściej opracowywane przez zainteresowane resorty np.: Ministerstwo Przemysłu Ciężkiego, Maszynowego, Łączności itd. a następnie zatwierdzane przez Ministerstwo Oświaty i Wychowania.

Aktualnie wyróżniamy trzy typy szkół kształcące specjalistów dla informatyki:

— **pięcioletnie technikum** na podbudowie 8-mio klasowej szkoły podstawowej
— **trzyletnie technikum** na podbudowie 3-letniej zasadniczej szkoły zawodowej
— **dwuletnie policealne studium zawodowe** na podbudowie liceum ogólnokształcącego.

Powyższe typy obowiązują w kształceniu młodzieży i dorosłych. Najwięcej wątpliwości budzi celowość kształcenia w pięcioletnim technikum, do którego przychodzą absolwenci szkół podstawowych o niezbyt jeszcze sprecyzowanych zamiłowaniach do swojego przyszłego zawodu i pracy. Natomiast najbardziej celowe jest chyba kształcenie w tzw. 2-letnich policealnych studiach zawodowych.

W wymienionych wyżej typach szkół obowiązują również trzy typy programów nauczania. Prezentowanie wszystkich tzw. siatek przedmiotów i godzin dla poszczególnych specjalności nie miałoby celu w tym artykule. Wspomnę jednak jakie obowiązują specjalności w dwuletnich policealnych studiach zawodowych:

- maszyny księgujące i fakturujące
- przetwarzanie danych
- programowanie elektronicznych maszyn cyfrowych
- urządzenia zewnętrzne komputera i urządzenia przygotowania danych
- elektroniczne maszyny matematyczne
- teletransmisja
- telekomutacja
- elektryczna i elektroniczna automatyka przemysłowa
- miernictwo elektryczne i elektroniczne
- elektronika
- elektromechanika ogólna.

Wydaje się, że obowiązujące specjalności są trochę rozproszone, a jednocześnie brak specjalności np.: dotyczącej technologii budowy maszyn cyfrowych oraz technologii budowy urządzeń przygotowania danych czy urządzeń peryferyjnych.

Wspólnym zarządzeniem (Nr 70/Org/74 z dnia 21.08.1974 r.) Ministerstwa Nauki, Szkolnictwa Wyższego i Techniki oraz Ministerstwa Oświaty i Wychowana, powołany został „zespół d/s opracowania projektu kształcenia i doskonalenia

PLAN NAUCZANIA
POLICEALNEGO STUDIUM ZAWODOWEGO

Zawód: technik operator sprzętu informatycznego
Specjalność: obsługa operatorska sprzętu informatycznego
Wiek: od 19 lat; okres nauczania: 1,5 roku
Podbudowa: liceum ogólnokształcące

Lp.	Przedmiot	Liczba godzin tygodniowo		
		I	II	III
1	Ekonomika i organizacja przedsiębiorstw oraz ośrodków obliczeniowych	4	4	4
2	Zasady księgowości i sprawozdawczości	4	4	2
3	Podstawy informatyki	2	2	—
4	Technologia przetwarzania oraz organizacja biblioteki taśmowej i dyskowej i przygotowania obliczeń	4	4	4
5	Operatorka na urządzeniach do przetwarzania danych — zaj. prakt.	2	2	2
6	Operatorka na EMC wraz z systemami operacyjnymi — zaj. prakt.	6	6	6
7	Zasady projektowania systemów EPD	2	2	—
8	Zastosowania EMC	—	2	2
9	Programowanie w językach symbolicznych	2	2	2
10	Ekonomia polityczna	2	2	2
11	Podstawy psychologii, socjologii i organizacji pracy	—	—	4
12	Rysunek techniczny	2	2	—
13	Maszynopisanie	2	2	2
14	BHP	2	—	—
15	Powszechna samoobrona	—	—	2
16	WF	2	2	2
17	Praktyka zawodowa*)			
	Razem:	36	36	36

*) w ośrodku obliczeniowym

PLAN NAUCZANIA
POLICEALNEGO STUDIUM ZAWODOWEGO

Zawód: technik programowania
Specjalność: programowanie EMC
Wiek: od 19 lat; okres nauczania 2 lata
Podbudowa: liceum ogólnokształcące

Lp.	Przedmiot	Liczba godzin tygodniowo			
		I	II	III	IV
1	Elementy matematyki wyższej i metod numer.	4	4	4	4
2	Podstawy informatyki	2	2	—	—
3	Programowanie w językach podstawowych	4	4	—	—
4	Programowanie w językach symbolicznych	8	8	4	4
5	Programowanie w językach algorytmicznych	—	—	10	10
6	Zasady projektowania systemów EPD	2	2	4	4
7	Zastosowanie EMC	—	—	2	2
8	Konserwacja oprogramowania i operowania systemami operacyjnymi	—	—	4	4
9	Podstawy nauk politycznych	2	2	—	—
10	Podstawy psychologii, socjologii i organizacji pracy	2	2	—	—
11	Ekonomika i organizacja przedsiębiorstw i ośrodków obliczeniowych	4	4	—	—
12	Język angielski	2	2	4	4
13	Rysunek techniczny	2	2	—	—
14	BHP	2	2	—	—
15	Powszechna samoobrona	—	—	2	2
16	Praktyka zawodowa*)				
17	WF	2	2	2	2
	Razem:	36	36	36	36

*) Praktyka zawodowa odbywać się będzie w ośrodkach obliczeniowych

kadr dla informatyki do 1980 r.". Zespół ten działał w oparciu o wytyczne zawarte w decyzji Prezydium Rządu Nr 3/74 z dnia 11.01.1974 r., a opracowany program miał być zaakceptowany przez kolejne posiedzenie Prezydium Rządu i wdrożony do realizacji.

Resort oświaty i wychowania miał w opracowywanym programie swój znaczny udział. Problemy, które zostały uwzględnione w programie, to:

- kształcenie informatyków na średnim poziomie
- kształcenie nauczycieli, przyszłych wykładowców przedmiotu „podstawy informatyki"
- kształcenie młodzieży szkół średnich ogólnokształcących w zakresie „podstaw informatyki"
- wyposażenie szkół w niezbędne pomoce dydaktyczne i sprzęt informatyczny.

Prace nad przygotowaniem omawianego wyżej programu trwały trzy miesiące i zakończyły się opracowaniem **„kompleksowego pragramu kształcenia i doskonalenia kadr dla informatyki do 1980 r."**, który na grudniowym (1974 r.) posiedzeniu Prezydium Rządu został zaakceptowany do wdrożenia.

Program określa zasadnicze kierunki kształcenia, natomiast nie precyzuje dokładnie wszystkich środków, metod czy form kształcenia. Będą one opracowywane sukcesywnie i wdrażane poprzez zainteresowane jednostki administracji państwowej.

W zakresie kształcenia średniej kadry informatyków w w/w programie sprecyzowano jedynie poszczególne specjalności. W miejsce dotychczasowych specjalności proponuje się więc wprowadzić następujące:

- obsługa operatorska sprzętu informatycznego
- programowanie elektronicznych maszyn cyfrowych
- instalowanie i konserwacja sprzętu elektromechanicznego (urządzenia przygotowania danych oraz sprzęt małej, średniej i dużej mechanizacji)
- instalowanie i konserwacja elektronicznych maszyn cyfrowych
- instalowanie i konserwacja urządzeń teleinformatyki
- technologia budowy elektronicznych maszyn cyfrowych
- technologia budowy urządzeń przygotowania danych.

Ponadto zaproponowano dwie specjalności nieinformatyczne, ale ściśle związane z informatyką, a mianowicie:

- elektronika ogólna, która winna objąć takie dziedziny jak: elektronika ogólna, miernictwo elektryczne i elektroniczne oraz automatykę przemysłową
- mechanika ogólna, która winna objąć takie dziedziny jak: mechanika ogólna, mechanika precyzyjna, mechanika aparatury kontrolno-pomiarowej.

Łącznie określono **siedem** specjalności informatycznych i **dwie** specjalności współdziałające z informatyką. Resort oświaty i wychowania proponuje dotychczasowe trzy typy

PLAN NAUCZANIA
POLICEALNEGO STUDIUM ZAWODOWEGO

Zawód: technik konserwator sprzętu informatycznego
Specjalność: instalacja i konserwacja urządzeń przygotowania danych oraz sprzętu małej, średniej i dużej mechanizacji
Wiek: od 19 lat; okres nauczania: 2,5 roku
Podbudowa: liceum ogólnokształcące

Lp.	Przedmiot	a Liczba godzin tygodniowo				
		I	II	III	IV	V
1	Podstawy informatyki	2	2	—	—	—
2	Podstawy elektrotechniki	6	6	—	—	—
3	Elektronika i automatyka przemysłowa	—	—	3	3	—
4	Miernictwo elektryczne i elektroniczne	2	2	2	2	—
5	Maszyny i napęd elektryczny	4	4	—	—	—
6	Elementy mechanizmów precyzyjnych	4	4	—	—	—
7	Aparaty i urządzenia elektryczne	2	2	2	2	2
8	Przyrządy i mechanika precyzyjna	2	2	3	3	—
9	Maszyny małej mechanizacji	—	2	4	4	4
10	Maszyny średniej mechanizacji	—	—	4	4	8
11	Maszyny dużej mechanizacji	—	—	4	6	8
12	Urządzenia przygotowania danych	—	—	8	8	14
13	Podstawy nauk politycznych	—	2	2	—	—
14	Podstawy psychologii, socjologii i org. pracy	2	2	—	—	—
15	Ekonomika i organ. przedsięb. i ośr. oblicz.	2	2	—	—	—
16	Rysunek techniczny	2	2	—	—	—
17	Materiałoznawstwo i technologia metali	4	4	—	—	—
18	BHP	2	—	—	—	—
19	Powszechna samoobrona	—	—	2	—	—
20	WF	2	2	2	2	2
21	Praktyka zawodowa*)					
		36	36	36	36	36

*) w ośrodku obliczeniowym

PLAN NAUCZANIA
POLICEALNEGO STUDIUM ZAWODOWEGO

Zawód: technik konserwator EMC
Specjalność: instalacja i konserwacja elektronicznych maszyn cyrowych
Wiek: od 19 lat; okres nauczania: 2,5 roku
Podbudowa: liceum ogólnokształcące

Lp.	Przedmiot	Liczba godzin tygodniowo				
		I	II	III	IV	V
1	Podstawy informatyki	2	2	—	—	—
2	Podstawy elektroniki	6	6	—	—	—
3	Podstawy automatyki	—	2	2	—	—
4	Podstawy elektroniki i układy elektroniczne	6	6	6	—	—
5	Przyrządy pomiarowe oraz miernictwo elektryczne i elektroniczne	4	4	4	—	—
6	Elementy mechaniczne w elektronice	4	—	—	—	—
7	Technologia metali i materiałoznawstwo	2	2	—	—	—
8	Programowanie w języku podstawowym	—	—	4	4	—
9	Organizacja i zasady funkcjonowania EMC	—	—	4	4	4
10	Jednostka centralna	—	—	4	10	12
11	Urządzenia WE/WY	—	—	4	10	10
12	Pamięć zewnętrzna	—	—	4	6	8
13	Podstawy nauk politycznych	—	2	2	—	—
14	Podstawy psychologii, socjologii i organizacja pracy	2	2	—	—	—
15	Ekonomika i organizacja przedsięb. i ośrodków obliczeniowych	2	2	—	—	—
16	Rysunek techniczny	2	2	—	—	—
17	Materiałoznawstwo i technologia metali	2	2	—	—	—
18	BHP	2	—	—	—	—
19	Powszechna samoobrona	—	2	—	—	—
20	WF	2	2	2	2	2
21	Praktyka zawodowa*)					
		36	36	36	36	36

*) Praktyka zawod. w ośrodku obliczeniowym

szkół systematycznie likwidować, a pozostawić i rozwijać jedynie **policealne studia zowodowe** o czasie kształcenia od 1,5 roku do 2,5 lat w zależności od specjalności.

Dla pierwszych siedmiu specjalności zostały określone siatki przedmiotów i godzin.

Jest to pierwsza propozycja, którą chcielibyśmy poddać pod publiczną dyskusję. Dla dwu pozostałych specjalności tzw. nieinformatycznych, jeszcze nie sprecyzowano siatki przedmiotów i godzin.

PLAN NAUCZANIA
POLICEALNEGO STUDIUM ZAWODOWEGO

Zawód: techniki konserwator teleinformatyki
Specjalność: instalacja i konserwacja urządzeń teleinformatyki
Wiek: od 19 lat; okres nauczania: 2 lata
Podbudowa: liceum ogólnokształcące

Lp.	Przedmiot	Liczba godzin tygodniowo			
		I	II	III	IV
1	Podstawy informatyki	2	2	—	—
2	Elementy i konstrukcje urządzeń teletechnicznych	2	2	2	—
3	Podstawy elektrotechniki	8	8	—	—
4	Podstawy teletransmisji	2	6	4	—
5	Urządzenia teletransmisyjne i teletechniczne	—	—	8	12
6	Urządzenia teletransmisyjne i telegraficzne	—	—	6	6
7	Zarys linii radiowych	—	—	2	2
8	Zarys telekomutacji	—	—	4	4
9	Linie telekomunikacyjne	—	—	4	6
10	Urządzenia zasilające	2	2	—	—
11	Miernictwo elektryczne i teletransmisyjne	6	4	—	—
12	Elektronika i automatyka	—	—	2	2
13	Technologia metali i materiałoznawstwo	4	2	—	—
14	Podstawy nauk politycznych	—	—	2	2
15	Podstawy psychologii, socjologii i organizacji pracy	2	2	—	—
16	Ekonomika i organizacja przeds. i ośr. oblicz.	2	2	—	—
17	Rysunek techniczny	2	2	—	—
18	BHP	2	—	—	—
19	Powszechna samoobrona	—	2	—	—
20	WF	2	2	2	2
21	Praktyka zawodowa*)				
	Razem:	36	36	36	36

*) w ośrodku obliczeniowym i w urzędach pocztowych.

PLAN NAUCZANIA
POLICEALNEGO STUDIUM ZAWODOWEGO

Zawód: technik konstruktor EMC
Specjalność: technologia budowy EMC
Wiek: od 19 lat; okres nauczania: 2 lata
Podbudowa: liceum ogólnokształcące

Lp.	Przedmiot	Liczba godzin tygodniowo			
		I	II	III	IV
1	Podstawy informatyki	2	2	—	—
2	Podstawy elektrotechniki	4	4	—	—
3	Podstawy techniki impulsowej	4	4	—	—
4	Organizacja i zasady funkcjonowania EMC	4	4	—	—
5	Architektura logiczna EMC	—	—	3	3
6	Struktura logiczna, jednostek kontrolnych i pamięci operacyjnych	—	—	9	9
7	Zasady budowy urządzeń WE/WY	—	—	8	8
8	Zasady budowy pamięci zewnętrznej	—	—	4	4
9	Konstrukcja i technologia produkcji układów elektronicznych	—	—	4	4
10	Konstrukcja i technologia produkcji elementów mechanicznych	—	—	4	4
11	Systemy zasilania	4	4	—	—
12	Miernictwo elektryczne i elektroniczne	4	4	—	—
13	Technologia metali i materiałoznawstwo	4	4	—	—
14	Podstawy nauk politycznych	—	—	2	2
15	Podstawy psychologii i socjologii i organizacja pracy	2	2	—	—
16	Ekonomika i org. przeds. i ośr. oblicz.	2	2	—	—
17	Rysunek techniczny	2	2	—	—
18	BHP	2	—	—	—
19	Powszechna samoobrona	—	2	—	—
20	WF	2	2	2	2
21	Praktyka zawodowa*)				
	Razem:	36	36	36	36

*) w przedsiębiorstwie produkcyjnym np.: ELWRO, ERA, BŁONIE

PLAN NAUCZANIA
POLICEALNEGO STUDIUM ZAWODOWEGO

Zawód: technik konstruktor urządzeń przygotowania danych
Specjalność: technologia budowy urządzeń przygotowania danych
Wiek: od 19 lat; okres nauczania: 1,5 roku
Podbudowa: liceum ogólnokształcące

Lp.	Przedmiot	Liczba godzin tygodniowo		
		I	II	III
1	Podstawy informatyki	2	2	—
2	Podstawy elektrotechniki	4	2	—
3	Organizacja i budowa maszyn małej, średniej i dużej mechanizacji	—	2	4
4	Zasady budowy urządzeń do przygotowania danych	—	4	4
5	Technologia produkcji niektórych maszyn małej, średniej i dużej mechanizacji	—	6	8
6	Technologia produkcji urządzeń do przygotowania danych	—	6	8
7	Systemy zasilania	4	2	—
8	Mechanika precyzyjna	4	2	—
9	Miernictwo elektryczne i elektroniczne	4	—	—
10	Maszyny i napęd elektryczny	4	2	—
11	Technologia produkcji układów elektronicznych	—	—	6
12	Technologia metali i materiałoznawstwo	4	2	—
13	Podstawy nauk politycznych	—	2	—
14	Podstawy psychologii, socjologii i organizacji pracy	2	2	—
15	Ekonomika i organizacja przeds. i ośr. oblicz.	2	—	—
16	Rysunek techniczny	2	—	—
17	BHP	2	—	—
18	Powszechna samoobrona	—	—	2
19	WF	2	2	2
20	Praktyka zawodowa*)			
	Razem:	36	36	36

*) w przedsiębiorstwie produkcyjnym np.: ELWRO, ERA, BŁONIE.

Szkolenie projektantów systemów w firmie ICL

W kwietniu br. autor był uczestnikiem kursu projektowania systemów zorganizowanego przez firmę ICL. Był to jeden z kolejnych stałych kursów przeznaczonych dla klientów firmy.

Ponieważ zakres tego kursu, jego organizacja i program znacznie się różnią od zakresu, organizacji i programów kursów projektowania systemów organizowanych w kraju, w przekonaniu autora, celowe jest omówienie tego kursu na łamach krajowego czasopisma informatycznego. Celowość takiego omówienia wynika również z konieczności wprowadzania do programów kursów projektowania systemów zmian, które wynikają z rozwoju techniki komputerowej.

Przeznaczenie, cel i zakres kursu

Kurs projektowania systemów firmy ICL jest przeznaczony dla: praktykantów w dziedzinie analizy systemów, programistów — kandydatów na analityków systemów oraz kandydatów na projektantów i organizatorów wdrażania systemów.

Od uczestników kursu jest wymagana znajomość zasad organizacji przedsiębiorstwa oraz znajomość techniki komputerowej. Celem kursu jest nauczenie uczestników:

— projektowania schematów komputerowego procesu przetwarzania danych i opisu jego poszczególnych operacji

— projektowania schematów przetwarzania dla procedur ręcznych poprzedzających i następujących po procedurach komputerowych

— projektowania zbiorów danych, opisu ich organizacji i metod dostępu

— projektowania dokumentów wejściowych i wyjściowych

— projektowania procedur kontroli danych i dokumentów wyjściowych

— obliczenia czasu przetwarzania w zaprojektowanym systemie

— opracowywania harmonogramów wdrażania systemów oraz zapoznanie uczestników z oprogramowaniem komputerów serii ICL 1900.

Kurs projektowania systemów jest częścią sześciotygodniowego szkolenia zatwierdzonego przez National Computing Centre (NCC) [1]. Na całe szkolenie składają się: jednotygodniowy kurs technik komputerowych, dwutygodniowy kurs analizy systemów oraz trzytygodniowy kurs projektowania systemów. Odbycie całego szkolenia upoważnia do zdawania egzaminu przed komisją NCC.

Chciałby zwrócić uwagę, że celem kursu jest n a u c z a n i e, a nie zapoznanie z powyższymi tematami. Organizacja kursu i przyjęte metody dydaktyczne są konsekwencjami tak sformułowanego celu kursu.

Kurs obejmuje projektowanie systemu w odmiennym niż w Polsce zakresie, wynikającym z innej roli projektanta systemu. W rozumieniu ICL projektant systemu to człowiek (zwykle pracujący w zespole), którego zadaniem jest zaprojektowanie systemu na podstawie (i zgodnie z nimi) potrzeb informacyjnych instytucji, przedstawionych mu przez analityka systemu. Wykaz potrzeb informacyjnych jest wynikiem analizy systemu, a projektant otrzymuje go po uzgodnieniu jego zawartości z kierownictwem instytucji, dla której system jest projektowany.

Organizacja kursu

Kurs, którego byłem uczestnikiem nosił nr 76. Miał on, jak wszystkie kursy projektowania systemów organizowane przez ICL, charakter stacjonarny. Efektywny czas

[1]) centralny państwowy ośrodek badawczo-rozwojowy w dziedzinie zastosowań informatyki w Wielkiej Brytanii.

trwania kursu wynosił 14,5 dnia. Zajęcia odbywały się w ośrodku szkoleniowym ICL w Beaumont koło Londynu. Dzienny rozkład zajęć był następujący: 9.15 — rozpoczęcie zajęć, 10.30—10.45 — przerwa na kawę, 12.45—14.00 — przerwa na lunch, 15.30—15.45 — przerwa na herbatę, 19.00 — zakończenie zajęć. W piątki zajęcia trwały do godz. 15.30, a soboty były wolne od zajęć.
Kurs przewidywał 130 godzin lekcyjnych w podziale: 42 godz. — wykłady, 66 godz. — ćwiczenia (projektowanie) oraz 22 godz. — omawianie projektów wykonanych na ćwiczeniach.
Mimo, że program kursu przewidywał zajęcia do godz. 19-tej, bardzo często prace nad projektami przeciągały się do godzin późniejszych. Tę dodatkową pracochłonność oceniam na ok. 30 godz. lekcyjnych.
W kursie brało udział ośmiu uczestników. Zostali oni podzieleni na dwa zespoły projektujące, których zadaniem było wykonywanie, najpierw poszczególnych elementów projektu systemu, a w końcowej fazie kursu, przygotowanie całego projektu. Projektowanie elementów systemu rozpoczęło się drugiego dnia kursu.
Uczestnicy wykonywali projekty systemu sprzedaży i fakturowania dla firmy Industrial Accessory Supplies Lmited (IAS) dokładnie opisanej w materiałach szkoleniowych. W materiałach szkoleniowych był również podany zestaw potrzeb informacyjnych firmy IAS w zakresie objętym projektowaniem, który był podstawą do projektowania.
Role przedstawicieli IAS wyjaśniających wątpliwości i przyjmujących proponowane rozwiązania, pełnili wykładowcy. Oni również wraz z uczestnikami kursu oceniali projekty wykonane przez poszczególne zespoły projektujące.
Omawianie wykonanych elemetów systemu i dyskusja przyjętych rozwiązań odbywały się nazajutrz po ćwiczeniach. Na prezentację całego projektu był przeznaczony ostatni dzień kursu.
Dla umożliwienia szerokiej dyskusji projekty były wykonywane przez zespoły na astralonie lub na planszach.
Do dyspozycji uczestników kursu były: sala wykładowa wyposażona w tablicę rzutnik pisma i tablicę do plansz oraz dwa pokoje do pracy w zespołach.
Wykłady były bogato ilustrowane przezroczami wykonanymi na astralonie.
Prawie do każdego wykładu były przygotowane drukowane materiały dydaktyczne zawierające rozszerzoną treść wykładu, treść ćwiczeń i zadań oraz ich rozwiązania. Rozwiązania zadań i ćwiczeń były wręczane uczestnikom przy omawianiu rozwiązań własnych.

Program kursu

Program kursu obejmował 10 modułów tematycznych. Tematy poszczególnych modułów były następujące:

Moduł 1. Dokumentowanie systemu: wykłady — 8 godzin, ćwiczenia — 6 godzin.
Analiza i projektowanie systemu: pojęce systemu, funkcje projektowania, fazy projektowania.
Zestawienie potrzeb informacyjnych (jako podstawa dla projektu systemu).
Opis systemu zgodnie ze standardami ICL, opis zbiorów, opis procesu przetwarzania.
Opis programów.
Ćwiczenia — w ramach ćwiczeń — 5 godz. lekyjnych było poświęconych na zapoznanie się z opisem istniejącego systemu w IAS oraz zestawem potrzeb informacyjnych w systemie sprzedaży i fakturowania.

Moduł 2. Metody prezentacji — środki wizualne: wykłady — 1 godzina, ćwiczenia — 1 godzina.
W module tym omówiono szczegółowo cechy dobrego wykładu oraz użycie rzutnika pisma i tablicy do plansz, jak również zasady wykonywania przezroczy na astralonie i plansz.
Ćwiczenia polegały na opracowaniu i wygłoszeniu przez zespoły wykładów na tematy: korzyści wynikające z posiadania słonia, zalety trójnożnych rajstop. [2])
Wykłady te były wygłoszone przy użyciu przygotowanych przezroczy i plansz.

[2]) tego rodzaju sztuczne i paradoksalne tematy miały na celu wyrobienie u przyszłych projektantów zdolności przekonującego uzasadniania proponowanych rozwiązań.

Moduł 3. Projektowanie formularzy: wykłady — 4 godziny, ćwiczenia — 10 godzin.
Zasady ogólne projektowania formularzy.
Projektowanie dokumentów wyjściowych (tabulogramów).
Projektowanie dokumentów wejściowych: dokumentów obiegowych, kart dziurkowanych, taśmy papierowej.
Ćwiczenia — w ramach ćwiczeń były wykonywane projekty dokumentów wejściowych i wyjściowych dla IAS

Moduł 4. Niektóre problemy przygotowania i oceny systemu: wykłady — 8 godzin, ćwiczenia — 2 godziny.
Klasyfikacja i indeksowanie: projektowanie indeksów, zależności między klasyfikacją a indeksowaniem, rodzaje indeksów, cechy dobrych indeksów, konwersja indeksów, zapobieganie i wykrywanie błędów.
Kontrola — znaczenie i metody kontroli w procesie przetwarzania.
Wprowadzenie do transmisji danych: konieczność transmisji danych, środki techniczne, przetwarzanie na bieżąco, bezpieczeństwo, przykłady zastosowań.
Ocena systemu: czynniki systemu podlegające ocenie, efekty systemu, koszty systemu.
Ćwiczenia z problematyki kontroli w systemie.

Moduł 5. Oprogramowanie: wykłady — 3 godziny.
Dostęp bezpośredni.
Pakiety użytkowe programów.

Moduł 6. Tablice decyzyjne: wykłady — 1 godzina, ćwiczenia — 1 godzina.

Moduł 7. Taśma magnetyczna: wykłady 2 godziny, ćwiczenia — 1 godzina.
Przetwarzanie danych na taśmach magnetycznych, problemy bezpieczeństwa zbiorów.
Ćwiczenia — projektowanie zbiorów taśmowych dla IAS.

Moduł 8. Dostęp bezpośredni: wykłady 10 godzin, ćwiczenia — 12 godzin.
Format zbiorów.
Organizacja zbiorów.
Przepełnenia zbiorów.
Metody dostępu.
Reorganizacja zbiorów i problemy bezpieczeństwa.
Zasady projektowana zbiorów.
Ćwiczenia — projektowanie zbiorów dyskowych dla IAS.

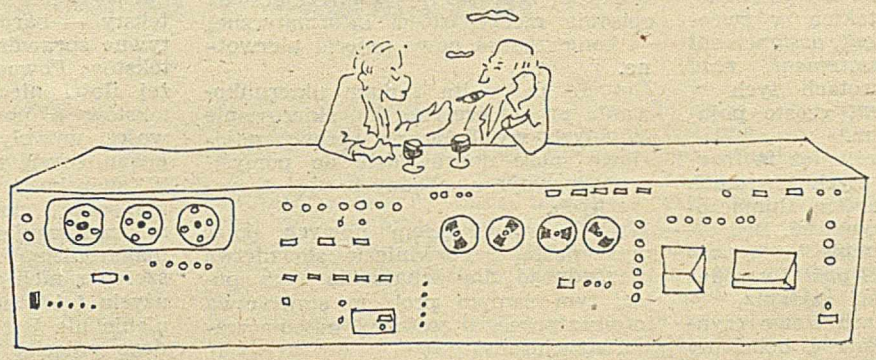
Moduł 9. Obliczanie czasu przetwarzania: wykłady — 2 godziny, ćwiczenia — 4 godziny.
Jednostka centralna.
Urządzenia peryferyjne.
Taśmy magnetyczne.
Dyski magnetyczne.
Ćwiczenia — obliczanie czasu przetwarzania w systemie zaprojektowanym dla IAS.

Moduł 10. Wdrażanie systemów: wykłady — 3 godziny, ćwiczenia — 1 godzina.
Metoda wdrażania systemów.
Metody planowania wdrażania: PERT, wykresy Gantta.
Ćwiczenia.
Pozostałe godziny lekcyjne ćwiczeń były poświęcone na wykonanie projektów, ich prezentację i dyskusję przyjętych rozwiązań.

Uwagi końcowe

Polskiego uczestnika kursu uderza przede wszystkim b. dobre przygotowanie kursu i duża intensywność szkolenia. Wykłady prowadzone metodą aktywną, z dużym udziałem krótkich ćwiczeń, zmuszały do brania bezpośredniego udziału w zajęciach, podnosząc w ten sposób efektywność szkolenia.
Zwraca również uwagę program kursu ograniczony do tematów bezpośrednio związanych z projektowaniem systemów bez np. historii komputerów, od której rozpoczyna się niemal każdy krajowy kurs z dziedziny informatyki. W programie kursu projektowania systemów ICL nie ma również wielu przedmiotów wykładanych na krajowych kursach projektowania systemów np. programowania. Natomiast duża liczba godzin przeznaczonych na ćwiczenia i dyskusję pozwala na wykonanie projektu systemu oraz na przedyskutowanie każdego jego elementu.
Chciałbym również podkreślić doskonałe warunki zakwaterowania w ośrodku ICL w Beaumont, różnorodność urządzeń rekreacyjnych oraz znakomitą kuchnię, którą b. wysoko oceniają nawet wstrzemięźliwi w zachwytach Anglicy.

Jerzy Choroszko



Kilka uwag na temat słownictwa informatycznego

Pojęcia określone w języku angielskim słowami (rzeczownikami) *file, record, statement* należą do ważniejszych pojęć informatyki i jest sprawą ważną znalezienie dla tych słów właściwych polskich odpowiedników — niestety jak dotychczas sprawy tej nie udało się rozwiązać w sposób zadawalający. W wypowiedzi niniejszej chciałbym zaproponować pewne nowe rozwiązania w powyższej sprawie.

Jako odpowiedniki słowa *file*, w dotychczasowej praktyce stosowane są przede wszystkim dwa słowa — **zbiór** oraz **plik**, ponadto czasami proponuje się zwrot **zbiór danych**.

Słowo **zbiór**, jakkolwiek najczęściej stosowane, jest absolutnie nie do przyjęcia z tego powodu, że znaczenie tego słowa w języku polskim jest raz na zawsze ustalone w matematycznej teorii zbiorów. Zaś zbiór w sensie teorii zbiorów jest czymś zupełnie innym niż *file*.

Takich jednoznacznych przeciwskazań nie ma w stosunku do słowa **plik**, jednak słowo to nie znalazło powszechnego uznania przez to, że budzi konkretne skojarzenia z plikiem kartek, który znowu nie jest czymś podobnym do *file*.

W stosunku do zwrotu **zbiór danych** można wysunąć obiekcje podobne jak w stosunku do słowa zbiór — zbiór danych jest to jakiś szczególny przypadek zbioru, zaś, jak to już powiedzieliśmy, *file* jest czymś zupełnie innym niż zbiór, a tym samym niż każdy szczególny przypadek zbioru.

Moim zdaniem pojęcie oznaczane słowem *file* jest pojęciem nowym, nie występuje w dotychczasowym języku polskim i dla jego oznaczenia konieczne jest wprowadzenie do języka nowego słowa — słowo to powinna być tak dobrane aby wywoływało skojarzenia zgodne z istotnymi właściwościami *file*. Postaramy się teraz zastanowić nad tymi właściwościami.

Przyjrzenie się temu jak zbudowana jest dowolna *file* pokazuje, iż jest to coś takiego w czym zawiera się zbiór pewnych elementów nazywanych najczęściej *records*. Pierwszym słowem narzucającym się dla oznaczenia *file* pod wpływem takich spostrzeżeń jest słowo zbiornik — jednak *file* jest w istocie pojęciem abstrakcyjnym, oderwanym, a tymczasem słowo zbiornik stosowane jest dla określenia pewnych bardzo konkretnych przedmiotów. Konieczne jest wobec tego znalezienie słowa nie obciążonego takimi konkretnymi skojarzeniami , a podobnego znaczeniowo do słowa zbiornik. Słowo takie udało mi się wynaleźć, a brzmi ono **ZBIORNIA** — proponuję to słowo jako odpowiednik *file*...

W drugiej części wypowiedzi postaram się omówić zagadnienie polskiego odpowiednika dla słowa *record*.

W dotychczasowej praktyce rozpowszechniły się dwa słowa odpowiadające temu słowu — **rekord** i **zapis**.

Słowo **rekord** budzi wątpliwości ze względu na swoje wyraźnie niepolskie brzmienie, zaś słowo **zapis** sprawia trudności ze względu na to, że oznacza też czynność i jego stosowanie prowadzi do konieczności posługiwania się zwrotem zapis zapisu, który nie budzi zachwytu.

Istnieje więc wyraźna potrzeba znalezienia nowego odpowiednika dla słowa *record*. Słowo takie może zostać wynalezione jedynie w oparciu o wyjaśnienie treści jaka jest w języku angielskim wkładana w słowo *record*. Wyjaśnienie to wymaga przeprowadzenia szerszego rozumowania — w pierwszej części wypowiedzi wprowadziliśmy słowo zbiornia i stwierdziliśmy, iż pojęcie oznaczane tym słowem ma ogólny charakter — teraz dodamy, że w informatyce mamy do czynienia z pewnymi szczególnymi realizacjami zbiorni. Realizacje te mają postać odpowiednio wyodrębnionych nośników napisów — elementy zawarte w takiej zbiorni przedstawiane są przez pewne wyodrębniające się od innych napisy — nazwać je można napisami elementarnymi lub podstawowymi. I właśnie taki napis elementarny czy podstawowy określany jest w języku angielskim słowem *record*.

Warto tu jeszcze zaznaczyć, iż użycie słowa *record* w języku angielskim nie jest konsekwentne — to co nazywa się zwykle *record* nalażałoby nazwać *elementary record*. Niektórzy badacze, np. prof. Borje Langefors, spostrzegli tę niekonsekwencję i zamiast słowem *record* posługują się zwrotem *elementary record*.

Tak więc jako odpowiednik słowa *record* lub ściślej zwrotu *elementary record* proponuję zwrot **NAPIS ELEMENTARY** lub **PODSTAWOWY**, w skrócie **E-NAPIS** lub **P-NAPIS**.

Zwrot **zapis e-napisu/p-napisu** nie budzi już wątpliwości — zwróćmy jeszcze uwagę, iż napis e-napisu jest w istocie wprowadzeniem elementu do zbiorni.

W trzeciej części wypowiedzi postaram się omówić zagadnienie polskiego odpowiednika dla słowa *statement* i równocześnie słowa *instruction*. Tłumaczenie tych słów sprawia stałe i znaczne trudności. Jako ich odpowiedniki stosowane są słowa: **zdanie, rozkaz, instrukcja** — odpowiedniość tych słów budzi jednak wątpliwości.

W informatyce słowo *statement* występuje w dwu użyciach. Do pierwszego użycia zaliczymy takich w zwrotach, jak: *go to statement, assignement statement, dummy statement, read statement* itp. W zwrotach tych w miejsce słowa *statement* często pojawia się słowo *instruction*.

Słowo *statement*, jak również *instruction*, występujące w tych zwrotach powinno być moim zdaniem, tłumaczone na zwrot **przepis czynności**. Będziemy wtedy mieli: **przepis czynności skoku, przepis czynności pustej czy pozornej, przepis czynności czytania.**

W niektórych wypadkach słowo czynność może być pomijana i będziemy wtedy mieli: **przepis podstawiania, przepis czytania** i np. **przepis pisania.**

Do drugiego użycia zaliczamy użycia słowa *statement* w zwrotach takich jak *DIMENSION statement, FORMAT statement*. W zwrotach tych słowo to powinno być tłumaczone jako **USTALENIE.**

Będziemy wtedy mieli **ustalenie wymiarowości, ustalenie formatu.**

Na zakończenie chciałbym jeszcze sformułować kilka ogólnych uwag na temat stanu słownictwa informatycznego. Po pierwsze chciałbym stwierdzić, iż przyczyny trudności z tym słownictwem są znacznie głębsze niż zwykle się sądzi, a tkwią one w całkowitym braku adekwatnych ujęć teoretycznych wszystkich zasadniczo zagadnień występujących w informatyce.

Z kolei przyczyną braku tych ujęć jest niewłaściwe ukierunkowanie prac teoretycznych — prace te winny być ukierunkowane na stworzenie całkowicie nowych pojęć przedstawiających składniki rzeczywistości informatycznej. Tymczasem w dotychczas prowadzonych pracach usiłuje się rzeczywistość informatyczna opisać za pomocą pojęć znanych, lub w najlepszym wypadku pojęcia informatyczne sprowadzić do już znanych pojęć, głównie matematycznej teorii zbiorów. Usiłowania te nie mają żadnych szans powodzenia, z tego prostego powodu, że znane pojęcia są niewystarczające do opisania rzeczywistości informatycznej — konieczne są nowe pojęcia pierwotne.

Dopiero całkowita zmiana ukierunkowania prac teoretycznych, skierowanie głównych wysiłków na sprawy pojęciowe może doprowadzić do pomyślnego rozwoju niezbędnych ujęć teoretycznych.

Wraz z rozwinięciem nowych pojęć musi nastąpić rozwinięcie słownictwa niezbędnego dla oznaczania tych pojęć, tym samym problem słownictwa informatycznego zostanie automatycznie rozwiązany.

Po drugie chciałbym zwrócić uwagę, iż w informatyce utrzymuje się bardzo, moim zdaniem, niebezpieczna skłonność do wytwarzania „makaronizmów XX-wieku" przez zaniechanie wynajdywania polskich słów dla oznaczania nowotworzonych pojęć i zadawalanie się bezpośrednim przejmowaniem słów obcych. Częstokroć postępowanie to oceniane jest jako właściwe i nawet korzystne, przytacza się na to nawet rozmaite argumenty, moim jednak zdaniem postępowanie to jest wysoce szkodliwe, co jest dość łatwo wykazać, podobnie jak łatwo jest wykazać niesłuszność argumentów mających rzekomo dowodzić korzyści z przejmowania obcych słów.

Przeprowadzimy teraz jedynie dowód szkodliwości. Źródłem szkodliwości jest to, że każde obce słowo leży poza czymś, co nazwałbym „więzami znaczeniowymi języka", dzięki którym każde słowo czysto polskie jest dość ściśle umiejscowione, zakorzenione wśród innych słów, co stwarza możliwość różnorodnego sprawdzania znaczenia słowa i wprowadza jasność co do tego znaczenia. Dla obcych słów takie umiejscowienie nie występuje, powoduje to silne rozmycie odczuć znaczenia tych słów, znaczenie obcych słów jest zawsze mocno niejasne. Z pomocą takich niejasnych, silnie rozmytych znaczeniowo słów można bezkarnie konstruować właściwie dowolne teksty — bardzo jest trudno o obiektywne sprawdziany sensowności takich tekstów. Powoduje to powstawanie dużej ilości nieokreślonych znaczeniowo tekstów o pozorach wysokiej fachowości, wywoływanych obfitym posługiwaniem się obcymi słowami.

W związku z powyższym wydaje się konieczne podjęcie przeciwdziałania rozprzestrzenianiu się współczesnych makaronizmów — oczywiste jest, że szereg z nich znajduje się w szerokim użyciu i ich natychmiastowe wykorzenienie nie jest możliwe. Lecz dla każdego pojęcia, dla którego istnieje już słowo obcego pochodzenia winno się wynaleźć i rozpowszechnić odpowiednie i dogodne słowo polskie. Jest dla mnie rzeczą nie budzącą wątpliwości, iż po jakimś czasie słowo takie, oczywiście o ile będzie odpowiedniej jakości, wyprze słowo obce.

Do słów, dla których należy wynaleźć czysto polskie odpowiedniki możliwe szybko zaliczyłbym słowa takie jak: system, struktura, informacja, program, algorytm, procedura, operacja, instrukcja, moduł, segment, monitor. Zabieg ten byłby bardzo korzystny również z tego względu, że wymienione słowa są z reguły wieloznaczne i praca nad wynajdywaniem polskich odpowiedników musiałaby doprowadzić do rozróżnienia poszczególnych znaczeń każdego z wymienionych słów i stworzenia, dla każdego z tych znaczeń, odrębnego słowa polskiego. Przyczyniłoby się to do znacznego uściślenia języka informatyki.

Tomasz Rawiński

XX-lecie czasopisma „Pomiary, Automatyka, Kontrola"

W lipcu br. upłynęło 20 lat od chwili ukazania się pierwszego numeru miesięcznika naukowo-technicznego Wydawnictw Czasopism Technicznych NOT „Pomiary, Automatyka, Kontrola" — organu Polskiego Komitetu Pomiarów i Automatyki NOT oraz sekcji Metrologii, Automatyki i Mechaniki Precyzyjnej SIMP.

Pismo ma charakter wielobranżowy i jest jedynym polskim periodykiem poświęconym następującym zagadnieniom:

— metrologii metodom i układom pomiarowym oraz konstrukcji aparatury pomiarowej przeznaczonej zarówno do celów przemysłowych, jak i naukowo-badawczych;

— teorii i praktyce automatycznego sterowania procesami technologicznymi oraz budowie aparatury regulacyjnej, a także zagadnieniom ruchowym;

— mechanice precyzyjnej w zakresie konstrukcji oraz technologii.

Pismo osiągnęło obecnie nakład 6700 egzemplarzy i obejmuje swym zasięgiem inżynierów, techników, studentów oraz pracowników naukowo-badawczych instytutów i wyższych uczelni.

Z okazji Jubileuszu zespół redakcyjny INFORMATYKI składa serdeczne gratulacje oraz życzy bratniemu czasopismu nowych sukcesów i dalszego rozwoju

Dokąd zmierzasz polska teleinformatyko?

Uwagi na temat konferencji naukowo-technicznej „Sieci teleinformatyczne" Gdańsk 5—7.06.1975 r.

Celem konferencji, która zorganizowana została przez Instytut Łączności i SEP przy współudziale CYFRONETU, IMM-u, IOK-u, OBRI, PKAPI, WAT-u i WIŁ-u było zaprezentowanie i przedyskutowanie dorobku w zakresie badań i zastosowań teleinformatyki oraz wymiana doświadczeń między specjalistami telekomunikacji i informatyki.
Konferencja podzelona była na sesje, które poświęcone były następującym zagadnieniom:

- sieciom teleinformatycznym
- systemom i usługom teleinformatycznym w Polsce
- transmisji, sterowaniu i komutacji
- zamierzeniom przemysłu, badaniom naukowym i kadrom.

Obrady konferencji otworzył prof. dr E. Kowalczyk — Minister Łączności referatem „Teleinformatyka wyrazem nowoczesnej organizacji procesów informatycznych". W sesji inauguracyjnej wygłoszono jeszcze następujące referaty:
— Kierunki rozwoju informatyki w Polsce — prof. dr hab. inż. A. Straszak prof. dr hab. inż. J. Kulikowski
— Kierunki rozwoju telekomunikacji w Polsce — doc. dr inż. W. Majewski i doc dr inż. A Zieliński
Nie trzeba nikogo przekonywać, że teleinformatyka wywalczyła sobie od dawna prawo do życia nie tylko jako słowo, lecz także jako dziedzina wiedzy, obejmująca dyscypliny nauki i techniki zajmujące się połączonymi zagadnieniami zdalnego zbierania, przesyłania, przechowywania informacji, sposobami ich reprezentacji, przetwarzania, jak również budową (konstrukcja i technologia) urządzeń maszyn i systemów służących do tych celów.
Autorzy referatów definiowali również teleinformatykę jako dziedzinę telekomunikacji zajmującej się zdalnym łączeniem urządzeń informatyki. Stwierdzili również, że teleinformatyka jest konsekwencją rozwoju informatyki i warunkuje jej nowoczesny rozwój.
Należy tu dodać, że resort łączności jako jedyny został upoważniony do koordynowania prac w zakresie teleinformatyki.
Rozwój tej dziedziny nauki w/g referatu „Kierunki rozwoju telekomunikacji w Polsce" można podzielić na 3 etapy:
— etap 1 — wykorzystanie w teleinformatyce łączy telefonicznych lub telegraficznych otwartych lub komutowanych. Tak więc sieć teleinformatyczna jest podzbiorem obecnej sieci telekomunikacyjnej, a praca sieci odbywa się w systemie *on* i *off line*.
etap 2 — realizacja wydzielonych sieci teleinformatycznych na bazie jednolitej sieci telekomunikacyjnej. W skład sieci wchodziłyby takie urządzenia teleinformatyki jak łącza, węzły komutacji itp.
Cechą charakterystyczną drugiego etapu jest tworzenie powszechnej sieci analogicznej do sieci telefonicznej czy teleksowej.
etap 3 — realizacja powszechnej sieci cyfrowej przeznaczonej dla wszystkich rodzajów służb telekomunikacyjnych (także dla teleinformatyki), w oparciu o rozwój komutacji elektronicznej (system E-10).
Jest dla każdego sprawą oczywistą, że teleinformatyka w Polsce zaczyna wchodzić w etap I a perspektywy na przyszłość nie są wcale optymistyczne. Niektóre zamierzenia resortu łączności w zakresie rozwoju telekomunikacji są bowiem następujące:
— nastąpi wzrost ilości aparatów telefonicznych na 100 mieszkańców z około 7 w r. 1975 do 12 w r. 1980, 20 — w roku 1985 i ok. 30 w r. 1990,
— nastąpi stopniowy rozwój sieci teleksowej i telefonicznej
— nastąpi stopniowe udostępnianie usług teleinformatycznych.
Należy tu podkreślić, że cechą obecnie istniejącej sieci telekomunikacyjnej w kraju jest ze względów technicznych jej niedostosowanie do potrzeb teleinformatyki. Dopóki bazą teleinformatyki będzie sieć telefoniczna, jakość usług będzie niska.
Stąd konieczność budowy takiej sieci, która umożliwiałaby przesyłanie danych z dowolną standardową szybkością, wynikającą z rzeczywistych potrzeb użytkownika przy zminimalizowanych nakładach finansowych. Względy ekonomiczne i niektóre udogodnienia dla użytkowników wskazują na celowość korzystania z sieci typu komutowanego, jednak istniejące sieci komutowane, jak już wspomniano wyżej, do tego celu się nie nadają. Wynika stąd wniosek, że przyszła sieć dla potrzeb teleinformatyki powinna być siecią komutowaną spełniającą wszelkie niezbędne warunki techniczne. Wstępna koncepcja takiej sieci określona kryptonimem KASTOR opracowana została przez Instytut Łączności. Oprócz łączy komutowanych sieć ta oferować będzie również użytkownikom łącza dzierżawione, a w przyszłości stać się jednym z ogniw światowej sieci teleinformatycznej.
Zagadnienia związane z siecią KASTOR omawiał doc. mgr inż. Janusz Sochacki w swym referacie pt. „Wstępna koncepcja rozwoju sieci teleinformatycznej w Polsce".
Ciekawy referat wygłoszony został przez prof. dr hab. inż. J. Kulikowskiego na temat: „Program rozwoju zastosowań informatyki a wymagania dla rozwoju krajowej sieci transmisji danych". Autor referatu przedstawił prognozy rozwoju teleinformatyki w kraju aż do roku 2000, a nawet dalej. Wydaje się, że cel jaki miała za zadanie konferencja został spełniony, jednak nasuwają się tu pewne refleksje.
W większości przypadków autorzy przewidując bujliwy rozwój sieci teleinformatycznej prezentują koncepcje oderwanie od rzeczywistości. Podają orientacyjne liczby węzłów i stacji abonenckich w r. 1980, 1990, 2000, nie opierając się na realnym zapotrzebowaniu. Jest sprawą oczywistą, że prognozowanie w tym zakresie jest niezwykle trudne i dlatego plany i zamierzenia powinny być dostosowywane do aktualnych potrzeb, możliwości z wyprzedzeniem najwyżej kilkuletnim. Należy tu z całą stanowczością i odpowiedzialnością stwierdzić, że sieci teleinformatyczne mogą być tworzone na określonym etapie rozwoju zastosowań systemów użytkowych. Dowodem na to twierdzenie może być fakt zakupu urządzeń i uruchomienia magistrali teleinformatycznej Gdańsk-Warszawa-Katowice, której brak jakichkolwiek aspektów użytkowych.
Czy 10 lat temu ktokolwiek mógł przewidywać i przewidział jak dziś będzie wyglądała polska informatyka i jak będzie wyglądała za lat 10?
Kolejnym elementem, którego brak w prognozach to rachunek ekonomiczny. Czy rzeczywiście budowa ogólnokrajowej sieci teleinformatycznej przyniesie określone efekty ekonomiczne, czy sieć ta będzie miała dostatecznie dużą liczbę użytkowników?
Są to pytania retoryczne, na które trzeba znać odpowiedź dziś, żeby prawidłowo prognozować na przyszłość.
W świetle referatu, jaki został wygłoszony przez mgr inż. Marka Wajcena ze Zjednoczenia MERA, w najbliższym czasie polski przemysł komputerowy wytwarzał będzie 4 rodzaje minikomputerów, komputery średniej wielkości JS oraz ODRA 1300, pamięci dyskowe, taśmowe i kasetowe drukarki wierszowe i znakowe oraz terminale dla systemów teleinformatycznych.
Zestaw środków technicznych przyszłych systemów przetwarzania Jednolitego Systemu przedstawia się następująco:
— programowany lokalny multipleksor PMPX

— zdalny multipleksor
— punkt abonencki AP 14
— programowany punkt abonencki DPA
— dialogowy punkt abonencki DPA
— niezbędne oprogramowanie telekomunikacyjne.

Zestaw. technicznych środków systemu teleprzetwarzania dla ODRY 1300 przedstawia się następująco:
— multipleksor lakalny MPX-325
— adapter liniowy telefoniczny UPD 305 8/5 do MPX-325
— adapter liniowy telefoniczny UPD 305-Y do MPX-325
— adapter liniowy telefoniczny UPD 305-X do tematu znakowego
— dialogowy programowany punkt abonencki
— oprogramowanie telekomunikacyjne.

Należy z uznaniem podkreślić, że zamierzenia, w tym względzie są ambitne, a ich zrealizowanie umożliwi tworzenie dużych systemów abonenckich w oparciu o różne typy maszyn.

W referatach prezentowanych na konferencji, traktujących o przyszłości polskiej teleinformatyki, brak jest informacji w jaki sposób resort łączności pragnie wykorzystać dostępny w kraju sprzęt i jakie są jego zamierzenia w tym zakresie.

W niektórych organizacjach potrzeby w zakresie teleinformatyki występują już obecnie Istnieje też kilka systemów teleinformatycznych o charakterze użytkowym chciażby prezentawany na konferencji system POLRAX 2 w ZETO Wrocław. Materiały konferencyjne w nieznacznym stopniu udzielają rad i wskazówek dla projektantów i technologów ośrodków obliczeniowych wykorzystujących transmisję danych, które powstają i będą powstawały w najbliższej przyszłości. Poza tym konferencja była z definicji konferencją naukowo-techniczną. Część naukowa konferencji została zrealizowana, zbyt mało natomiast posiadała elementów techniczno-praktycznych, nie wliczając oczywiście do tego wycieczek do ośrodków obliczeniowych. Końcowym elementem konferencji były tezy do dyskusji, które po niewielkich poprawkach zastały zaakceptowane przez konferencję. Moim zdaniem jednak zespół opracowujący tezy — kompleksowo, ale zbyt ogólnie potraktował całe zagadnienie.

TELEINFORMATYK

Wnioski konferencji „Sieci teleinformatyczne"

1.

W rozwoju sieci teleinformatycznych obowiązują następujące zasady:
— budowa sieci jest warunkiem koniecznym prawidłowego — przestrzennego rozwoju informatyki,
— planowanie rozwoju sieci odbywa się przy harmonijnym współdziałaniu informatyki i telekomunikacji,
— celowe jest budowanie z odpowiednim wyprzedzeniem eksperymentalnych i pilotowych sieci transmisji danych.

2.

Ze względu na konieczność zapewnienia niezbędnej jakości transmisji, rozwój sieci teleinformatycznych wymaga znacznych nakładów finansowych na rozwój i modernizację sieci telekomunikacyjnej. Nakłady te są większe, niż by to wynikało z prostego przewidywania obciążeń ruchowych sieci telekomunikacyjnej.

3.

Obserwowane trendy rozwoju światowych sieci teleinformatycznych wskazują na współistnienie w perspektywie czasu:
— komutowanych sieci teleksowych i telefonicznych wykorzystywanych dla potrzeb teleinformatyki,
— jedno- i wielokomputerowych systemów teleinformatycznych wykorzystujących dzierżawione łącza telekomunikacyjne i procesory komunikacyjne,
— ogólnodostępnej sieci teleinformatycznej zbudowanej na bazie wydzielonej sieci transmisji danych; jest to rozwiązanie najbardziej efektywne z punktu widzenia potrzeb społecznych i kosztów ekonomicznych.

4.

Potrzeby w zakresie usług teleinformatycznych mogą być zaspokojone poprzez budowę szeregu wydzielonych sieci teleinformatycznych, z których każda wykorzystuje wyłącznie dla swoich celów określone elementy (np. łącza) sieci telekomunikacyjnej. Rozwiązania takie mogą być dobrze dopasowane do charakteru usług teleinformatycznych świadczonych przez te sieci, aczkolwiek prowadzi to do nieekonomicznego wykorzystania elementów sieci telekomunikacyjnej. Ogólnodostępna sieć teleinformatyczna powinna zapewnić pełniejsze wykorzystanie elementów sieci telekomunikacyjnej, co jest rozwiązaniem bardziej racjonalnym. Przyszła sieć teleinformatyczna powszechnego użytku powinna zapewnić możliwie szeroki zakres usług informatycznych bardzo wysokiej jakości.

5.

Niezbędne jest zachowanie harmonijnego związku między sukcesywną rozbudową sieci teleinformatycznej, a rozwojem przemysłowego potencjału produkcyjnego w zakresie komutacyjnego i transmisyjnego sprzętu teleinformatycznego.

Refleksje nad stanem programowania

Uważny czytelnik INFORMATYKI zapewne pamięta, że gdy na łamach Nr 3 z 1974 r. prezentowaliśmy gdyńskie ZETO, sporo uwagi poświęciliśmy grupie osób pracującej nad „rozpakowaniem" firmowego systemu operacyjnego ICL-GEORGE-3 przeznaczonego do maszyn ICL 1900 i ODRA-1305. I oto w stosunkowo niedługim okresie (biorąc pod uwagę, że było to nie jedyne zadanie programistów) prace zostały uwieńczone pełnym sukcesem: system został wdrożony, a zasady jego funkcjonowania zostały wyłożone na seminarium, które z inicjatywy ZETO zostało zorganizowane w dn. 30 maja w Gdyni.

Osiągnięcia zespołu z pionu software'owego mgr Jarosława Ruckiego, pracującego pod wodzą mgr inż. Romana Jelińskiego nie da się zakwestionować. I jeżeli w dalszej części wywodów, ktoś się może poczuć dotknięty to z pewnością nie programiści z Gdyni. To co im zlecono wykonali dobrze.

Gdyńskie seminarium dało mi asumpt do rozważań znacznie ogólniejszych, a ich przedmiotem są tak sprawy związane z GEORGE-3 jak i samą zasadą realizacji prac software'owych. Zacznijmy od GEORGE. Seminarium unaoczniło, że pojawił się kolejny argument podkopujący zasadność koncepcji kupna angielskiej licencji; za tą decyzją przemawiać miał walor doskonałego oprogramowania maszyn ICL 1900 — tymczasem okres rozpakowania oprogramowania podstawowego wydłużył się mocno. Nie mówiąc już o tym, że kosztowało to sporo (około 19 mln złotych) a jednak i tak można odnieść wrażenie, że z wdrożeniem GEORGE nieco się pośpieszono. Brzmi to oczywiście paradoksalnie, zwłaszcza w sytuacji gdy większość ośrodków (a napewno Zakłady Elektronicznej Techniki Obliczeniowej) uskarża się na zwiększenie ilości zamówień użytkowników ponad dotychczasowe możliwości sprzętu. Paradoks jest tylko pozorny. GEORGE-3 — dobry system operacyjny nie wiele tu pomoże, albowiem: podstawowa konfiguracja ODRY 1305 właściwie nie stwarza możliwości stosowania tego systemu operacyjnego (za mało pojemna pamięć operacyjna, słabe wyposażenie w urządzenia peryferyjne) a z rozbudową konfiguracji kłopoty są znane. Wiadomo także, że efektywność GEORGE-3 rośnie przy pracy w zdalnym dostępie, zaś maleje przy pracach wsadowych, tymczasem na powszechność pracy poprzez końcówki — szanse są odległe.

Trudno mieć absolutną pewność, czy aby właśnie ośrodki zdecydowanie usługowe (a takim jest ZETO) były najbardziej predystynowane do wyręczania producenta; istotnie tak ośrodek gdyński jak i wrocławski (pracujący równolegle nad wdrożeniem GE-

ORGE) widziały w tym konieczny interes — dysponując maszynami ICL względnie ODRA, ale pamiętać należy, że do tak poważnego zadania desygnować musiały bodaj najlepszą kadrę programistów, co odbić się musiało na wykonywaniu podstawowych zobowiązań ZETO.

Można mniemać, że swego czasu popełniono spory błąd rezygnując z powołania specjalnej grupy do zaadaptowania pakietów firmowych ICL dla wszystkich użytkowników komputerów tej firmy, a jeżeli już nie to, bodaj zdobyć się na koordynacje prac. Uniknęłoby się zbędnych wydatków jakie stworzyła sytuacja powstała z racji prac nad GEORGE, a mianowicie równoczesnej pracy nad jednym zadaniem przez OE GUS z jednej strony a ZETO Gdynia i Wrocław — z drugiej (nie mówiąc już o COGiE Katowice — ale to inna sprawa).

Korzyść z wdrożenia systemu operacyjnego GEORGE-3 nie musi mieć charakteru partykularnego. Środki na ten cel zostały przyznane z budżetu problemu węzłowego bez zobowiązania instytucji, które je wydatkowały do rozpowszechniania systemu wśród innych posiadaczy komputerów ICL/ODRA. Wprawdzie w Gdyni słyszałem, że nikomu pomocy się nie odmówi, ale zapewne tylko w ramach wolnego czasu.

Tyle o refleksjach nad sprawą GEORGE-3, które mogłyby być aktualnie także dla perypetii z niejednym systemem. Właściwie perypetie te mają wspólny mianownik, któremu na imię: brak koordynacji, brak kontroli.

Na wielu konferencjach i seminariach jak złe echo powraca stale wołanie o informację i współpracę programistów z różnych ośrodków. Robi się nawet nieśmiałe próby stworzenia platform wymiany w postaci giełd systemów powielarnych, ofert współpracy i... wszystko toczy się po staremu. Zwyczajną stała się sytuacja równoległych acz oddzielnych prac nad uruchomieniem bądź adaptacją tych samych systemów. Co ośrodek — to inny system planowania i rozliczania gospodarki materiałowej, płac i zatrudnienia, organizacji produkcji itp. Pomnaża to kwoty wydatkowane na opracowywanie systemów, absorbuje myśl i pracę ludzi, blokuje czas pracy komputerów.

Zwyczajnym też staje się brak amortyzacji środków zaangażowanych w opracowywanie systemów (lub „rozpakowywanie"), ponieważ nie opracowuje się ich do końca. Trudno tu ubolewać nad iluzoryczną szansą uruchomienia dużych systemów, choć sporo kosztowały.

Ale trudno zrozumieć dlaczego przerwano prace nad adaptacją obcych systemów firmowych takich jak PROMPT (ICL) czy BOMP (IBM), w które to prace zainwestowano przecież nie małe środki.

Może nieco trywializując, należy chyba jednak powiedzieć, że w sytuacji gdy próbujemy już ustalić na czym będziemy liczyć, czas na ustalenie — jak będziemy liczyć. Co ciekawsze świadomość faktu staje się już niemal powszechna. Pora na koordynację przynajmniej dużych przedsięwzięć w dziedzinie oprogramowania jest już najwyższa.

Wypada mi się tu zastrzec, że nie czuję się kompetentny udzielać rad kto i jak powinien koordynować i kontrolować. Nie znaczy to bynajmniej bym czuł się zwolniony z obowiązku gdy widząc duży i niesłychanie aktualny problem nie postarałbym się go przynajmniej zasygnalizować.

Krystyn Bernatowicz

ZE ŚWIATA

HONYWELL-BULL dokonuje fuzji z CII

W kwietniowym numerze „Informatyki" przedstawiliśmy główne problemy związane z niepewną przyszłością jedynego całkowicie francuskiego producenta maszyn cyfrowych — firmy CAMPAGNIE INTERNATIONALE pour L'INFORMATIQUE (C.I.I.) Trudności panujące na światowym rynku sprzętu informatycznego oraz wewnętrzne tarcia pomiędzy głównymi akcjonariuszami C.I.I. — grupą THOMSON, a COMPAGNIE GENERALE D'ELECTRICITE — zmusiły rząd francuski, na którego barkach spoczywało praktycznie w całości finansowanie prac badawczo-rozwojowych prowadzonych przez C.I.I. do bezpośredniej interwencji i doprowadzenia do radykalnej zmiany organizacyjnego podporządkowania „pupilka Francuzów".

Warunki porozumienia

Wielomiesięczne, zakulisowe pertraktacje, w których główną rolę przejęło Ministerstwo Przemysłu i Badań, doprowadziły wreszcie do ostatecznego rozwiązania. Michel d'Ornano — szef tego Ministerstwa — ogłosił 12 maja komunikat, będący wynikiem obrad powołanej i kierowanej bezpośrednio przez prezydenta Republiki Walerego Giscard d'Estainga Rady Międzyministerialnej, w którym powiadomiono o połączeniu handlowego i badawczo-rozwojowego wydziału C.I.I. ze spółką HONEYWELL-BULL, filią amerykańskiej kampanii elektronicznej HONEYWELL Inc. Warunkiem zaintegrowania działalności obu spółek jest odstąpienie przez firmę ze Stanów Zjednoczonych 19% kapitału zakładowego HONEYWELL-BULL zarządowi francuskiemu oraz grupie C.G.E. tym samym udział kapitału francuskiego w nowym ugrupowaniu przemysłowo-handlowym C.I.I.-HONEYWELL-BULL wyniesie ogółem 53%, gdyż 34% udziałów pozostanie w gestii Compagnie des MACHINES BULL, holdingu uczestniczącego finansowo w firmie HONEYWELL-BULL. Chociaż koncern amerykański posiadać będzie tylko 47% akcji nowo utworzonego organizmu, to jednak będąc największym akcjonariuszem będzie miał w praktyce najwięcej do powiedzenia przy podejmowaniu decyzji ekonomicznych.

Działalność produkcyjna dawnej C.I.I., obejmująca zakłady w Tuluzie oraz w miejscowości Les Andelys na północ od Paryża, a także wydziały badawczo-rozwojowe zajmujące się militarnym zastosowaniem produkowanych urządzeń, należeć będą do odrębnego przedsiębiorstwa, które zostanie utworzone i będzie zarządzane przez grupę THOMSON C.S.F. Poważne, na razie bliżej nieokreślone, udziały w tym przedsiębiorstwie posiadać będą organy rządowe.

Jednym z najważniejszych zagadnień, wokół którego toczyły się pertraktacje opóźniające podjęcie ostatecznego porozumienia, była wysokość kredytów przeznaczonych na finansowanie działalności badawczo-rozwojowej, których udzielić miał rząd francuski nowemu ugrupowaniu.

Koncern HONEYWELL Inc., będąc sam w poważnych tarapatach finansowych, żądał początkowo sumy dwóch miliar-

dów franków, by ostatecznie wynegocjować 1.200 milionów niskoprocentowego kredytu, który udostępniony będzie ugrupowaniu w przeciągu czterech najbliższych lat. Jednocześnie HONEYWELL zapewnił sobie praktyczną wyłączność na dostawę maszyn cyfrowych na potrzeby administracji państwowej.

Czy koniec snu o „europejskiej" informatyce?

W ten sposób, decyzje podjęte przez rząd francuski oznaczają początek końca procesu tworzenia niezależnej od firm amerykańskich produkcyjnej bazy „europejskiej" lub co najmniej francuskiej informatyki.
Utworzona w grudniu 1966 roku pod auspicjami samego Generała de Gaulle'a COMPAGNIE INTERNATIONALE pour l'INFORMATIGUE, będąca wówczas symbolem dążenia do uniezależnienia się francuskiego przemysłu od dominacji Stanów Zjednoczonych w dziedzinach najnowocześniejszej technologii, uległa nieubłaganym prawom rządzącym gospodarką kapitalistyczną, w której największe nawet ambicje narodowe nie mogą same utrzymać przy życiu przyczółków w froncie opanowanym przez giganty zza Atlantyku.

Niepowodzenia w realizacji Planu CALCUL, a zwłaszcza w zakresie programu ogólnoeuropejskiej współpracy w ramach UNIDATY (ugrupowania powstałego w lipcu 1973 roku w celu kooperacji przy produkcji i sprzedaży maszyn cyfrowych, skupiającego C.I.I., niemieckiego SIEMENSA oraz holenderskiego PHILIPSA) przypominały podobne kłopoty związane z konstrukcją francusko-brytyjskiego ponaddźwiękowego samolotu pasażerskiego „Concorde". W obu przypadkach doskonałość rozwiązań technologicznych nie wystarczyła do osiągnięcia sukcesu handlowego, a możliwość współpracy europejskiej okazała się iluzoryczna. W odróżnieniu jednak od losu programu „Concorde", który kosztował już oba zainteresowane rządy zbyt drogo, by można było się z niego wycofać, UNIDATA pozwala jeszcze na stosunkowo „honorowe" zwinięcie żagli. Już w ubiegłym roku SIEMENS rozpoczął rozmowy z UNIVAC — oddziałem informatycznym amerykańskiego koncernu SPERRY RAND — w sprawie zacieśnienia współpracy w zakresie rozwoju wspólnych badań i produkcji urządzeń informatycznych.

Umowa pomiędzy C.I.I. a HONEYWELL oznacza wycofanie się z programu zachodnioeuropejskiej współpracy jego najbardziej zagorzałego zwolennika. Chociaż w komunikacie ministra d'Ornano zawarty jest punkt, mówiący o konieczności poszukiwania nowych rozwiązań współpracy europejskiej, lecz chyba nawet sami autorzy tekstu nie wierzą zbytnio w realność takiej możliwości. Jak wiadomo, jedynie rząd francuski próbował uczynić z problemu stworzenia europejskiego przemysłu informatycznego zagadnienie bardziej polityczne niż gospodarcze, nie znajdując jednakże zrozumienia u swoich partnerów niemieckich i holenderskich. Nowa ekipa rządowa Walerego Giscard d'Estainga, reprezentująca „nowy styl" kierowania sprawami gospodarczymi, szybko zdecydowała się na zrezygnowanie z ambitnych, lecz nad wyraz kosztownych prestiżowych programów badawczo-produkcyjnych, realizowanych bez pomocy amerykańskich gigantów.

Partnerzy w porozumieniu

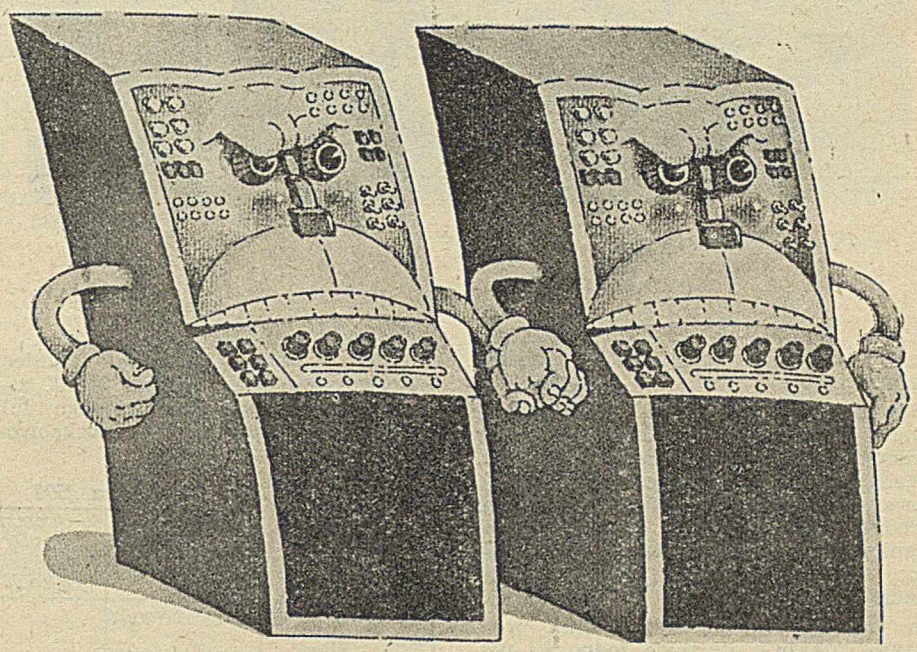
W oficjalnych wypowiedziach przedstawicieli władz francuskich oraz kierownictwa grupy C.G.E., której prezydent Ambroise Roux, jak się okazało, był jednym z głównych rzeczników połączenia C.I.I. z HONEYWELL-BULL, akcentuje się, iż podjęta ostatnio decyzja oznacza zwiększenie bezpośredniego wpływu czynników francuskich na rozwój całości krajowego przemysłu informatycznego. Czy większościowy udział kapitałów francuskich w kapitale zakładowym nowego ugrupowania oraz formalna kontrola państwowych instytucji bankowych nad wykorzystywaniem udzielonych kredytów równoznaczna jest z możliwością faktycznego ukształtowania przyszłości tego sektora?
Porównajmy podstawowe dane liczbowe, decydujące o sile ekonomicznej spółek uczestniczących w prozumieniu. Koncern HONEYWELL Inc., drugi pod względem wielkości produkcji koncern zajmujący się informatyką na świecie, zatrudniał pod koniec ubiegłego roku ponad 90 tysięcy pracowników i osiągnął ogólne obroty w wysokości 2,6 miliarda dolarów (około 10,5 miliarda franków francuskich), przy czym jego wydział informatyczny „Honeywell Information Systems", w którym 18,5% udziałów posiada inny amerykański koncern GENERAL ELECTRIC, osiągnął w 1974 roku obroty w wysokości 1,2 miliarda dolarów.
Francuska filia — HONEYWELL-BULL zatrudnia obecnie 15.700 osób, w tym we Francji pracuje 10.500 pracowników. Jej roczne obroty wynoszą 534 miliony dolarów (2,2 miliarda franków), z których osiągnięto zysk w wysokości 25 milionów dolarów.
Natomiast C.I.I. osiągnęła w ubiegłym roku obrót wartości 1.380 milionów franków, przy zatrudnieniu 8,8 tysiąca osób, przy czym spółka przez ostatnie trzy lata nie zanotowała nadwyżki wpływów nad kosztami, a pogłębiający się deficyt pokrywany był rządowymi subwencjami (ich wysokość globalna w ciągu ośmiu lat istnienia C.I.I. wyniosła 1,5 miliarda franków).
Z prostego porównania tych najbardziej ogólnych liczb wynika, iż znaczenie działalności na rynku francuskim sprzetu informatyki jest dla koncernu HONEYWELL stosunkowo niewielkie, gdyż ogranicza się wyłącznie właśnie do sprzedaży urządzeń i świadczenia usług informatycznych, która to dziedzina nie przynosi ostatnio HONEYWELLOWI zbyt dużych zysków. Spowodowane jest to trwającymi od 1970 roku trudnościami związanymi z koordynacją programu produkcyjnego tej firmy z przyjętymi przez nią wydziałami informatycznymi koncernu GENERAL ELECTRIC (między innymi, w maju 1970 roku GENERAL ELECTRIC odstąpił HONEYWELLOWI swe udziały w firmie BULL). W łonie kierownictwa HONEYWELL nie ma jednomyślności na temat dalszego kontynuowania działalności w zakresie produkcji maszyn cyfrowych i możliwym jest, iż w przypadku zaostrzenia się trudności w zbycie tego sprzętu na rynku Stanów Zjednoczonych, HONEYWELL mógłby zrezygnować z utrzymywania tej produkcji i odstąpić część swoich wydziałów produkcyjnych innej firmie, przy czym za ewentualnego kontrahenta wymienia się koncern CONTROL DATA CORPORATION, z którym HONEYWELL podpisał w kwietniu bieżącego roku porozumienie w sprawie współpracy naukowo-technicznej.
W przypadku takiego obrotu sprawy, rząd francuski zmuszony byłby do podjęcia od nowa sprawy organizacyjnego podporządkowania całego ugrupowania C.I.I.-BULL. Charakterystycznym jest tutaj fakt zawarcia w porozumieniu między francuskim Ministerstwem Przemysłu i Badań a HONEYWELL klauzuli o możliwości przejęcia przez rząd francuski w określonych okolicznościach większościowego udziału w nowo powstałej grupie.
Dlaczego więc, biorąc pod uwagę niezbyt zdrową sytuację finansową koncernu HONEYWELL, rząd francuski wybrał właśnie takiego partnera dla rozwoju swego przemysłu sprzętu informatyki? Wydaje się, iż po prostu nie istniała inna alternatywa, gdyż żaden z amerykańskich konkurentów HONEYWELLA nie zgodziłby się na współpracę z europejskimi instytucjami rządowymi. Uzgadnianie wspólnego stanowiska w takich związkach jest o wiele bardziej trudniejsze niż przy porozumieniach pomiędzy przedsiębiorstwami, w których czynniki polityczne nie odgrywają tak istotnej roli.
Z drugiej strony, rząd francuski nie może się obecnie zdecydować na zrezygnowanie z bezpośredniego udziału w krajowym przemyśle sprzętu informatyki. Jednym z ważnych powodów, dla których wybrano właśnie koncern HONEYWELL jest oczywiście jego obecność na rynku francuskim, co ułatwi stosunkowo szybkie przegrupowanie, a przecież przeciągająca się niepewność co do przyszłości C.I.I. najbardziej wpłynęła na obecne trudności przedsiębiorstwa.
Za wcześnie jeszcze oczywiście na ocenę ewentualnego porozumienia z HONEYWELL na konkurencyjność francuskiego sprzętu informatycznego na rynkach światowych. Argumenty zwolenników współpracy z Amerykanami o możliwości wkroczenia z produkcją ugrupowania C.I.I.-BULL na rynek Stanów Zjednoczonych za pośrednictwem sieci dystrybucyjno-serwisowej HONEYWELLA są raczej mocno przesadzone, przynajmniej na najbliższe lata, niezbędne na opracowanie wspólnej gamy maszyn cyfrowych.
Bardziej realną możliwością jest włączenie sprzętu produkowanego przez dawna C.I.I. do sieci dystrybucyjnej amerykańskiego partnera istniejącej w Europie i rozprowadzającej dotychczas produkcję jego niemieckiej i włoskiej filii. Jest to tym ważniejsze, że SIEMENS odmówił przed kilkoma miesiącami udostępnienia swej sieci hand-

lowej dla maszyn cyfrowych serii Iris.
Odrębnym zagadnieniem jest możliwość zdyskontowania przez nawe ugrupowanie dotychczasowych sukcesów C.I.I. na rynku krajów socjalistycznych (umowy kooperacyjne z węgierskim Videotonem w zakresie produkcji minikomputerów serii „1010" oraz urządzeń peryferyjnych, dostawy do Rumunii, Bułgarii oraz współpraca z przemysłem kubańskim).

Niepokój związków zawodowych

Ogłoszenie o decyzji połączenia COMPAGNIE INTERNATIONALE pour l'INFORMATIQUE z HONEYWELL-BULL wywołało natychmiast gwałtowne reakcje organizacji związkowych w zakładach C.I.I. w Tuluzie i Les Andelys.
Najliczniejsza organizacja związkowa C.G.T., zorganizowała 13 maja kilkugodzinny strajk protestacyjny oraz wielką manifestację, w czasie której przedstawiciele załogi raz jeszcze podkreślili, że ich zdaniem jedynym rozwiązaniem, zapewniającym pewną przyszłość francuskiego przemysłu informatycznego jest pełna nacjonalizacja C.I.I. jak i HONEYWELL-BULL.
Zakładowa organizacja Francuskiej Partii Komunistycznej oświadczyła, iż rządowa decyzja oznacza włączenie przodujących pod względem technologicznym zakładów francuskich w orbitę wielkich amerykańskich koncernów międzynarodowych i oddania w ręce dyrektorów zza Atlantyku decydującego głosu w sprawach produkcji i zatrudnienia. Podobne oświadczenie złożyła zakładowa komórka Partii Socjalistycznej P.S.U.
W dalszym ciągu przypomina się masowe zwolnienia dokonane przez GENERAL ELECTRIC po przyjęciu kontroli nad COMPAGNIE des MACHINES BULL. Oświadczenia o konieczności zachowania obecnego stanu zatrudnienia w nowym ugrupowaniu, zawarte w komunikacie ministra d'Ornano nie mogą uspokić pracowników, gdyż podobne zapewnienia udzielano w kwietniu 1964 roku, w kilka miesięcy przed redukcją personelu w zakładach BULL.
Stanowisko instancji związkowych poparł prezydent C.I.I. Michel Barre, składając jednocześnie Radzie Administracyjnej przedsiębiorstwa rezygnację z zajmowanego stanowiska. Prezydent Barre należał do najzagorzalszych zwolenników zwiększenia udziałów rządowych w kapitale C.I.I. oraz zacieśnienia współpracy zachodnioeuropejskiej w ramach UNIDATY, sugerując jednocześnie rozszerzenie tego ugrupowania o inne firmy, a zwłaszcza brytyjską I.C.L. i szwedzką DATASAAB. Jego pozycja była już od dawna zagrożona, szczególnie po rozwiązaniu w sierpniu ubiegłego roku Delegatury d/s Informatyki i oddelegowaniu jej przewodniczącego Maurica Allegre na odległą placówkę aż na Nowej Kaledonii.
Zmiana na najwyższym stanowisku w C.I.I. pociągnie za sobą z pewnością dalsze zmiany personalne w kierownictwie, co także nie przyczyni się do ułatwienia procesu koordynacji działalności obu elementów składowych nowej spółki.

Jerzy Miłek



Wersja dyskowa minikomputera System 1500

W drugiej połowie 1974 roku amerykańska firma Singer Business Machines zaprezentowała wersję dyskową minikomputera System 1500.

Artykuł podaje krótką charakterystykę ogólną wersji dyskowej Systemu 1500, przedstawia podstawową jednostkę zestawu — 1503, rodzaje jednostek dyskowych oraz omawia podstawowe konfiguracje minikomputera.

Charakterystyka ogólna

Widok ogólny minikomputera System 1500 w wersji dyskowej przedstawia rysunek 1.

Wymiary minikomputera: 132 cm — szerokość, 76 cm — głębokość, 90 — 106 cm — wysokość.

Waga podstawowego zestawu wynosi 181 kg. Napięcie zasilania: 208—258 V.

Temperatura pracy: 15,5°C—45°C a wilgotność względna: 20—80%.

W skład podstawowego zestawu wersji dyskowej minikomputera wchodzi:
— jednostka podstawowa 1503,
— jednostka pamięci dyskowej 1540 lub 1543.

System 1500 zachował swą modularną budowę, co pozwala na wybór odpowiedniej konfiguracji zestawu w zależności od potrzeb.

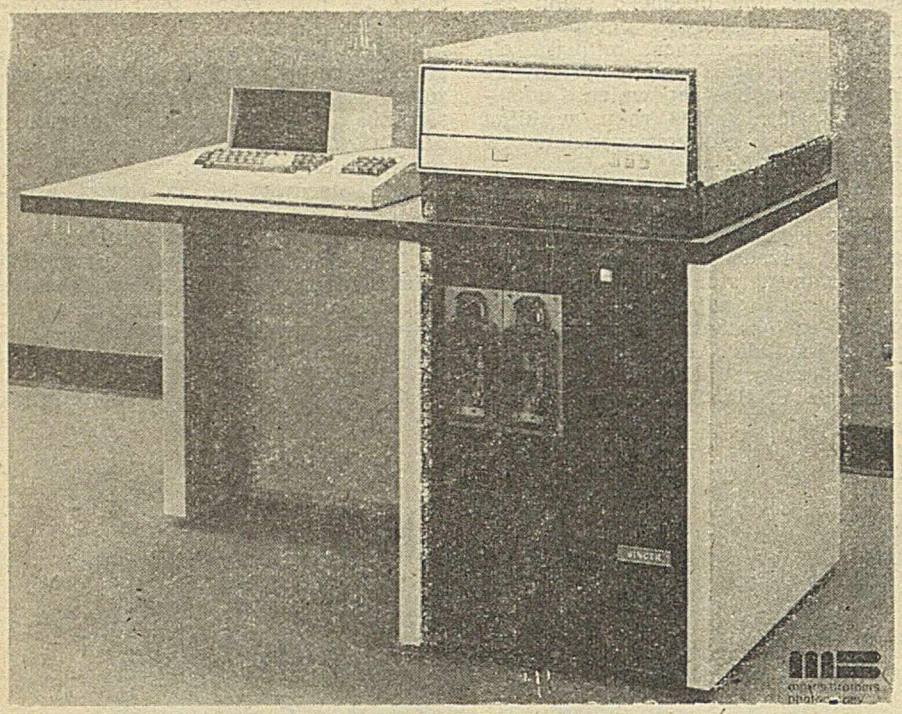
Dodanie pamięci dyskowej powoduje, że System 1500, który w swoich dotychczasowych wersjach służył głównie do wprowadzania danych, stał się minikomputerem uniwersalnym.

Charakterystyka jednostki 1503

W skład standardowego zestawu jednostki 1503 wchodzą:
— procesor (identyczny jak w jednostce 1501/1502),
— pamięć operacyjna o pojemności 16K (1K = 1024) bajtów,
— kanał dyskowy,
— dwie jednostki pamięci taśmowej kasetowej (1/8 cala),
— kanał szeregowy (dla lokalnych urządzeń zewnętrznych).

Pamięć operacyjna może być powiększona przez dodanie jednego lub dwóch modułów po 8K bajtów każdy.

Normalnie około 9K bajtów pamięci jest zajętych przez system operacyjny minikomputera. Pozostała część pa-



Tablica I. Charakterystyka jednostek pamięci dyskowej Systemu 1500

	Model 1540	Model 1543
liczba dysków stałych (jednostek dyskowych)	1	1
liczba dysków wymiennych (jednostek dyskowych)	0	1
ogólna liczba dysków (jednostek dyskowych)	1	1
maksymalna liczba jednostek dyskowych	4	4
maksymalna liczba dysków	4	4
liczba powierzchni roboczych/dysk	2	2
liczba ścieżek/powierzchnię	203	204
liczba sektorów/ścieżkę	24	24
liczba sektorów/dysk	9 744	9 792
liczba bajtów danych/sektor	256	256
liczba bajtów etykiety/sektor	12	12
liczba bajtów/dysk	2 494 464	2 506 752
liczba bajtów/jednostkę dyskową	2 494 464	5 013 504
maksymalna pojemność pamięci dyskowej w bajtach	9 977 856	20 054 016
przeciętny czas dostępu (ms)	70	38
szybkość obrotowa (obr./min)	1 500	1 500

mięci dostępna jest dla programów użytkownika.

Standardowo za pośrednictwem kanału dyskowego dołączona jest do maszyny jedna jednostka dysków. Kanał dyskowy maksymalnie steruje pracą czterech jednostek dyskowych. Podstawowe operacje dyskowe (czytanie, pisanie, poszukiwanie) odbywają się automatycznie bez udziału jednostki centralnej.

W standardowym wyposażeniu maszyny znajdują się dwie jednostki pamięci taśmowej (1/8 cala). Ponadto można dołączyć do jednostki 1503 jednostkę 1533, zawierającą dwie dodatkowe jednostki pamięci kasetowej. Kanał szeregowy pozwala dołączyć lokalnie jednostki standardowej pamięci taśmowej (szerokość 1/2 cala), drukarki, bądź jednostki 1501/1502. Poza tym można wyposażyć maszynę w synchroniczne lub asynchroniczne adaptery komunikacyjne, umożliwiające zdalną współpracę minikomputera w sieciach teleinformatycznych.

Pamięć dyskowa.

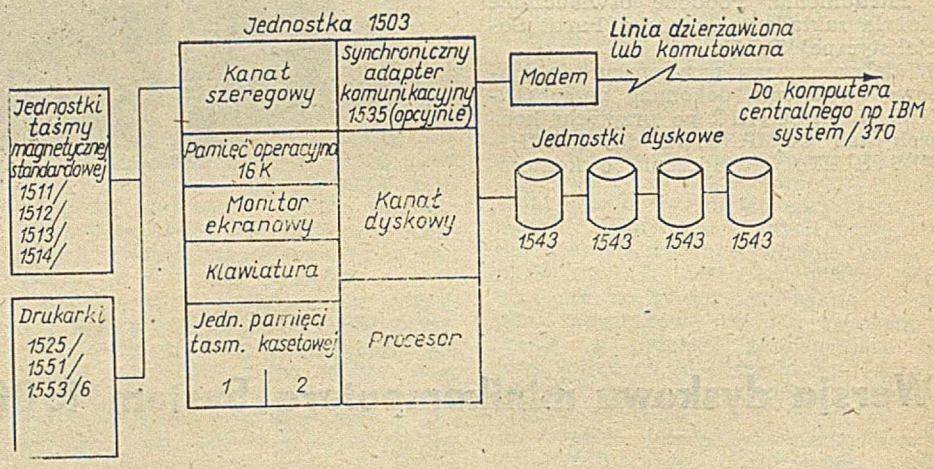
W Systemie 1500 używane są dwa typy jednostek pamięci dyskowej. Model 1540 zawiera dysk stały, natomiast jednostka 1543 posiada jeden dysk stały i jeden wymienny.

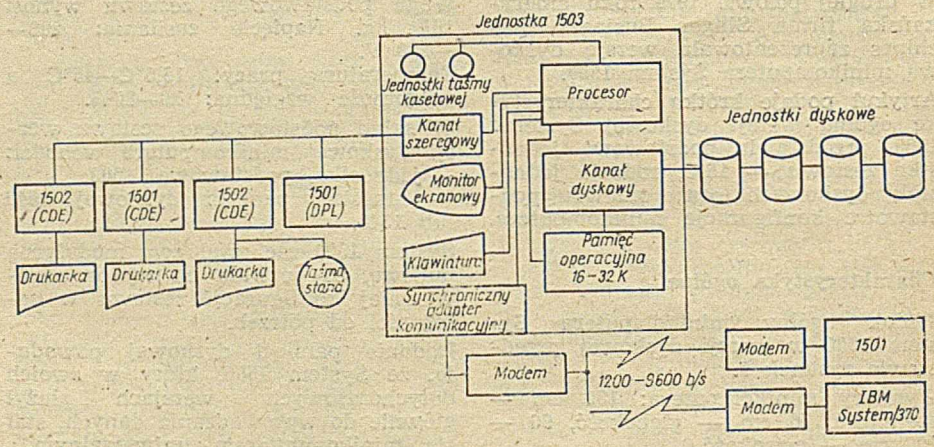
Charakterystyka pamięci dyskowej podana jest w tablicy 1.

Gęstość zapisu informacji na wewnętrznej ścieżce wynosi 2200 BPI. Szybkość transmisji w kanale dyskowym wynosi 195 000 bajtów. System sterowania operacjami dyskowymi — Data Base Mannager (DBM) zajmuje 8K bajtów pamięci operacyjnej. DBM pozwala operować na zbiorach dyskowych o dostępie sekwencyjnym, indeksowo-sekwencyjnym oraz bezpośrednim.

Aktualne oprogramowanie pamięci dyskowej pozwala czytać 8—10 zapisów ze zbioru sekwencyjnego lub 5—6 zapisów ze zbioru indeksowo-sekwencyjnego (przy założeniu że 1 zapis zawiera 256 bajtów).

Fizyczna długość sektora wynosi 263 bajtów. Pierwsze 12 bajtów każdego





sektora stanowi jego etykietę i nie są dostępne dla użytkownika.

Typowe konfiguracje minikomputera.

Istnieją dwie zasadnicze konfiguracje pracy minikomputera System 1500 w wersji dyskowej:

— układ pojedynczy (ang. Stand-Alone Mode)

— układ grupowy (ang. Clustered Mode).

Możliwe jest połączenie w/w układów w układ mieszany (ang. Combined Mode).

Aktualna wersja DBM steruje pracą każdego z wymienionych układów.

Układ pojedyńczy.

Schemat układu pojedyńczego przedstawia rys. 2.

Minikomputer pracuje w tym przypadku jako element sieci teleinformatycznej z możliwością komunikowania się poprzez binarne łącze synchroniczne na linii dzierżawionej lub komutowanej z komputerem centralnym, np. IBM System/370.

Ze względu na stosunkowo dużą, jak dla minikomputera pojemność pamięci dyskowej (20 Mbajtów), wzrosła zdecydowanie moc obliczeniowa minikomputera w porównaniu z konfiguracjami zbudowanymi w oparciu o jednostki 1501 lub 1502.

Pamięć dyskowa Systemu 1500 pozwala na operowanie na zbiorach o łącznej pojemności około 80 tysięcy rekordów po 256 bajtów każdy.

Pozwala to na przejmowanie części zadań w ramach sieci teleinformatycznej, a następnie przekazanie wyników obliczeń do komputera centralnego. Ponadto lokalne wykorzystanie minikomputera w układzie pojedynczym daleko przekracza fazę wprowadzania danych procesu przetwarzania, co jest główną funkcją jednostki 1501.

Za pośrednictwem kanału szeregowego można dołączyć do 64 jednostek pamięci taśmowej standardowej lub drukarek.

Układ grupowy.

Przykład konfiguracji minikomputera System 1500 w układzie grupowym przedstawia rysunek 3.

Poprzez kanał szeregowy do jednostki 1503 dołączone są w tym przypadku bezpośrednio jednostki 1501 lub 1502. Maksymalna liczba jednostek hierarchicznie podporządkowanych (1501/1502) pracująca w kanale szeregowym wynosi 32.

Z kolei do kanału szeregowego w minikomputerach niższego szczebla można dołączyć jednostki standardowej pamięci taśmowej bądź drukarki Systemu 1500.

Minikomputery niższego szczebla w wyniku połączenia z jednostką 1503 mają dostęp do informacji na dysku. Poszczególni użytkownicy minikomputerów 1501/1502 mogą więc przetwarzać wspólne lub oddzielne zbiory dyskowe.

Maksymalna odległość między minikomputerem 1503 a jednostką podrzędną 1501/1502 wynosi 457 metrów. Przykładem wykorzystania układu grupowego może być przedsiębiorstwo, gdzie w poszczególnych jego oddziałach (miejscach powstawania informacji wejściowych), zainstalowane są jednostki 1501/1502.

Poprzez synchroniczne łącze transmisji danych możliwa jest komunikacja z jednostkami 1501 zainstalowanymi w odległych oddziałach przedsiębiorstwa oraz z komputerem centralnym np. typu IBM System/370, zainstalowanym w biurze jednostki nadrzędnej. Przez połączenie układu pojedyńczego z układem grupowym otrzymuje się układ mieszany (ag. Combined Mode).

Władysław S. Gajewski

Kto potrzebuje programistów?*)

Liczbę programów, napisanych dla ponad 400 tys. komputerów zainstalowanych w chwili obecnej na świecie, szacuje się na około 400 mln. Blisko 90% stanowią duplikaty funkcjonalne. Tak wysoki udział duplikatów, świadczący o marnotrawstwie zasobów, wynika z następujących przyczyn:

- system szkolenia kadry programistów przewiduje naukę pisania programów, nie obejmując w dostatecznym stopniu tematyki wyszukiwania gotowych programów i ich adaptacji
- brak jest obecnie właściwej informacji o dotychczas napisanych programach
- większość ludzi schematycznie podchodzi do kwestii rozwiązywania problemów, nie będąc w stanie określić problemu, gdy dysponują jego gotowym rozwiązaniem.

Zróbmy założenie, że komputery są urządzeniami ogólnie dostępnymi i spełniają zadania „inteligentnych techników", a z usług komputerów korzystamy na zasadach współpracy z dobrym pracownikiem technicznym (np.: „p. J., proszę się tym zająć".)

Nie trzeba podkreślać, że założenie należy traktować jako przyszłościowe, gdyż opisana forma współpracy nie zawsze jest możliwa do zrealizowania nawet w zespole ludzkim. Tym nie mniej warto zastanowić się, czy nie jesteśmy w stanie przyśpieszyć momentu pojawienia się opisanej sytuacji. Wydaje się, że osiągnięcie pewnych efektów w tym kierunku umożliwiłaby realizacja kilku prostych zasad:

- źródłem informacji o programach powinien stać się projekt systemu (lub wynik tzw. analizy systemu), ponieważ dla większości osób program zawiera informację w formie nieczytelnej
- programy powinny być pisane w taki sposób, żeby oprócz ich autora, mogła z nich skorzystać każda osoba zainteresowana
- należy zapewnić warunki umożliwiające dostęp do informacji o istnieniu poszukiwanego programu (lub projektu systemu). Informację tę można zorganizować na zasadzie dopełnienia —tzn. sygnalizować brak programu (lub projektu systemu).

Podaną na wstępie szacukową liczbę napisanych programów, obliczono zakładając, że dla 1 komputera waha się ona w granicach 100÷1000. Przeciętny koszt napisania programu wynosi 100÷1000 dolarów i na tej podstawie koszt oprogramowania obecnie zainstalowanych komputerów można oszacować na 400 mld dolarów. Fakt, że około 90% wszystkich programów, stanowią duplikaty, pozwala ocenić liczbę programów niepowtarzalnych na około 4 mln. Można również przyjąć, że z tej liczby około 90% stanowić mogą programy, które z takich czy innych względów nie zainteresują innych użytkowników i w rezultacie otrzymamy szacunkową liczbę programów „podstawowych". Większość osób zainteresowanych nie jest w stanie uzyskać jakiejkolwiek pożytecznej, z ich punktu widzenia, informacji na temat poszukiwanego programu, a nieliczni mogą mieć dostęp do około 0,2% programów unikalnych. Oznacza to, że z każdych 10 000 napisanych programów zaledwie 2 mogą być dostępne dla osoby zainteresowanej.

Założenie, że 90% napisanych programów stanowią duplikaty może wydawać się zbyt pesymistyczne. Gdybyśmy przyjęli, że udział duplikatów jest niższy i kształtuje się w granicach 70%, to po przeprowadzeniu poprzednio omówionego toku rozumowania, uzyskalibyśmy katastrofalny wręcz obraz

*) opracowano na podstawie art. Gordona R. M.: Who needs programers, DATAMATION 1975, nr 2.

możliwości dostępu do pełnej informacji o programie. Okazałoby się mianowicie, że informacje o dwóch z 90 000 programów unikalnych mają szanse trafić do osób zaintersowanych. Obie omówione sytuacje w kategoriach pieniężnych oznaczają: w I przypadku straty 360 mld dolarów, w II przypadku — 280 mld dolarów.
Oznacza to, że w chwili obecnej tolerujemy, mało — stwarzamy warunki umożliwiające skandaliczne wręcz marnotrawstwo środków osobowych, rzeczowych i finansowych. Żaden z użytkowników komputerów nie jest bowiem skłonny przeznaczyć pieniędzy na sfinansowanie przedsięwzięcia, którego efektem byłoby stworzenie systemu rzetelnej i pełnej informacji w zakresie opracowanych programów i który dawałby odpowiedź na pytanie — czy to, co chcę zrobić, zostało uprzednio ujęte w formie programu? Na zakończenie artykułu autor cytuje wyjątek z raportu Mc Kinesey'a z 1969 roku [1]).
„Z punktu widzenia dochodowości oceniamy, że większość przedsięwzięć z zakresu informatyzacji, w prawie wszystkich badanych przypadkach była równoznaczna ze stratami. Szybszy, kosztowniejszy bardziej skomplikowany sprzęt, większe ilościowo i coraz wyższe wynagrodzenie personelu EPD, nieskończenie skomplikowane i prawie genialne zastosowania — takie przypadki obserwowaliśmy wszędzie. Coraz mniej z tych nowych zastosowań daje pozytywne rezultaty w sensie finansowym, co potwierdza prawo o zmniejszającym się stopniu zwrotu nakładów. Ogólnie można więc powiedzieć, że jak dotąd, potencjalna możliwość zyskownego zastosowania komputera została zaledwie „muśnięta". (is)

1) Unlocking the Computer's Profit Potential Mc Kinsey Co., Inc., New York, 1969.

Minikomputer ogólnego przeznaczenia jako procesor czołowy

Podstawowym zadaniem procesora czołowego (ang. *front end processor*) w strukturze systemów cyfrowych jest uwolnienie komputera głównego od wykonywania zrutynizowanych czynności związanych z następującymi funkcjami:

— sterowanie wolnymi urządzeniami we/wy
— zarządzanie zewnętrznymi pamięciami masowymi
— obsługa transmisji danych
— organizacja współpracy lokalnie połączonych komputerów roboczych.

Funkcje obsługi wolnych urządzeń peryferyjnych ze względu na to, że dane mają zwykle postać szeregową (taśma papierowa, karty perforowane, itp.) nie są skomplikowane. Systemy przerwań nowoczesnych komputerów zapewniają także minimalizację czasu zajętości procesora podczas wykonywania operacji we/wy. Wieloprogramowość umożliwia ciągłe wykorzystanie mocy obliczeniowej, mimo, że część programów oczekuje na skompletowanie zapisów danych.

Systemy cyfrowe oparte na sprzężonych komputerach roboczych są z kolei dość rzadko spotykane w typowych zastosowaniach techniki obliczeniowej. Z tych względów większość zastosowań procesorów czołowych dotyczy zarządzania pamięciami masowymi i sterowania transmisją danych. W tradycyjnych systemach cyfrowych funkcje logiczne związane z obsługą tych urządzeń zgrupowane były w procesorze centralnym, a urządzeniem wykonawczym była odpowiednia jednostka sterująca (rys. 1). W celu odciążenia procesora od części tych operacji logicznych rozbudowywano jednostki sterujące od strony układowej.

Duża różnorodność stosowanych terminali, szybkość, kodów i trybów transmisji z jednej strony oraz coraz bardziej wyrafinowane sposoby organizacji zbiorów w pamięciach masowych spowodowały, że budowa uniwersalnych jednostek sterujących sterowanych układowo jest bardzo kosztowna.
Dlatego niektórzy producenci maszyn cyfrowych rozpoczęli na przełomie lat sześćdziesiątych i siedemdziesiątych produkcję programowalnych jednostek sterujących, które były w praktyce wyspecjalizowanymi minikomputerami. Do konstrukcji takich można zaliczyć programowalną jednostkę sterującą PF56 pracującą w ICL-owskim podsystemie dyskowym ED3 30/60 i jednostkę 3704/3705 firmy IBM służącą jako procesor czołowy do zarządzania transmisją.
Ze względu na zniżkową tendencję cen minikomputerów ogólnego przeznaczenia produkowanych w dużych ilościach, w ostatnich latach rozpoczęto szerzej stosować te maszyny jako procesory czołowe także w standardowych zestawach obliczeniowych. Przykładem może tu być zastosowanie przez firmę ICL minikomutera Modular One COMPUTER TECHNOLOGY LTD do zarządzania bazą danych w podsystemie dyskowym EDS 100 oraz sterowania transmisją danych w podsystemie komunikacyjnym 7930.
Szczególne korzyści ekonomiczne przynosi zastosowanie minikomputerów ogólnego przeznaczenia jako procesorów czołowych dla komputerów produkowanych w krótkich seriach, nietypowych lub też pracujących z nietypowymi urządzeniami we/wy. Sytuacja taka stworzyła pole do działania dla stosunkowo małych firm, dysponujących niewielkim zapleczem sprzętowym, które przede wszystkim drogą rozbudowy oprogramowania minikomputera sprzęgają ze sobą sprzęt obliczeniowy różnych producentów.

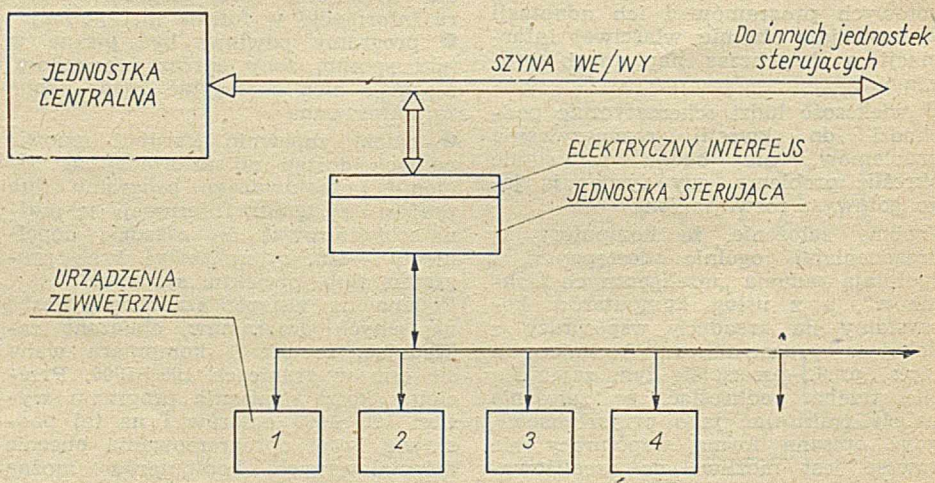
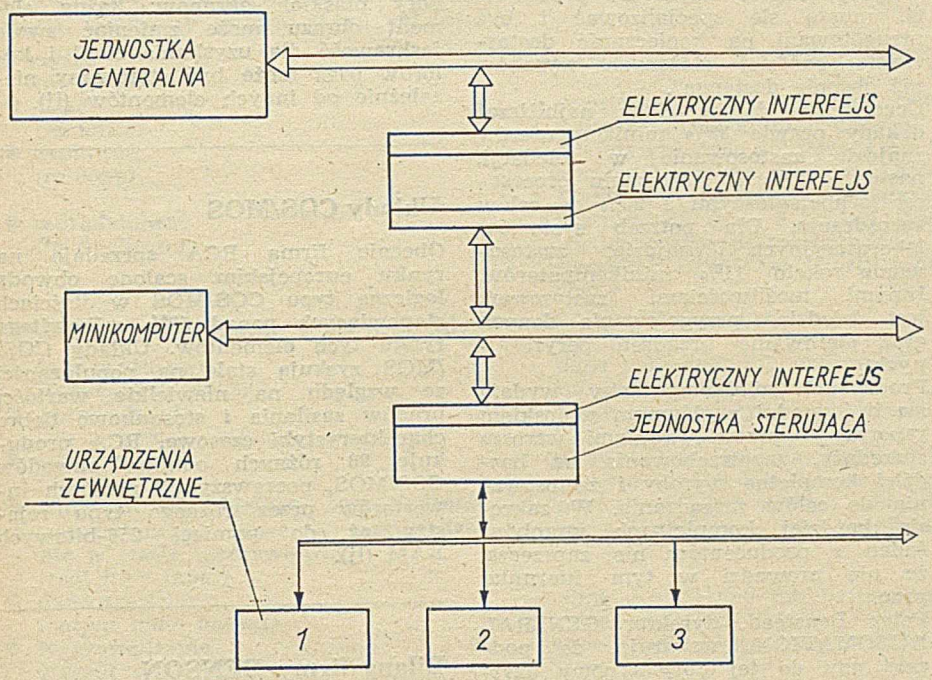


Tabela I. Podstawowe dane techniczne jednostki dyskowej TELEX 5312

Pojemność jednostki pamięci dyskowej	29 Mbajtów
czas dostępu:	
— minimalny	10 Ms
— średni	32 Ms
— maksymalny	60 Ms
szybkość transmisji	312 kbajtów/s

Tabela II. Podstawowe dane techniczne minikomputera MICRO 16

długość słowa	16 bitów
pamięć	4.64 kB
czas cyklu	950 ns
adresowanie bezp.	do 4 kB
max liczba urządzeń zewnętrznych	do 64
max szybkość transmisji	13,7 Mbajtów



Jedną z takich firm jest brytyjska firma TEKNOS, zatrudniająca kilkunastu pracowników, która opracowała metodę dołączenia systemu dyskowego firmy TELEX do komputerów MIŃSK 32 i ODRA 1305. System dyskowy TELEX składa się z jednostki sterującej 5328 oraz 1—9 jednostek pamięci dyskowej (30 Mbajtów) (tabela I). System ten jest kompatybilny z systemem IBM 2314. Ponieważ bezpośrednie połączenie jednostki sterującej do MIŃSKA 32 jest niemożliwe, firma zaproponowała wykonanie tego połączenia przy użyciu minikomputera MICRO 16 produkcji angielskiej firmy DIGICO (tabela II). Wybór tego minikomputera podyktowany został faktem, że oprócz spełnienia podstowowych wymagań technicznych charakteryzuje się on stosunkowo niską ceną.

Większość jednostek centralnych współpracujących z urządzeniami we/wy wyposażona jest w szynę we/wy (ang. *input-output bus*), która służy nie tylko do przesyłania danych w formie szeregowej lub równoległej, lecz także do przesyłania sygnałów kontrolnych, sterujących itp. do przyłączonej jednostki sterującej. Tak więc jednostka sterująca ma za zadanie:

— rozpakowanie danych przesyłanych przez szynę we/wy oraz przystosowanie ich do wymagań urządzenia zewnętrznego

— zapewnienie wszelkich niezbędnych sygnałów sterujących wymaganych przez urządzenie zewnętrzne.

Konsekwencją dużej różnicy szybkości przesyłania informacji między jednostką centralną, sterującą i urządzeniem zewnętrznym jest konieczność wyposażenia jednostki sterującej w rejestry buforowe o dużej pojemności, które praktycznie stanowią układy pamięci. Oprócz tego jednostka sterująca musi zawierać elektryczne układy interfejsu umożliwiające przyłączenie do szyny we/wy.

Zasygnalizowane wyżej problemy skłoniły do podjęcia prac nad uzupełnieniem jednostek sterujących uniwersalnym minikomputerem.

Rozwiązanie takie umożliwia:

— dołączenie dowolnego urządzenia we/wy do komputera głównego (mimo braku kompatybilności)

— wykorzystywanie standardowego oprogramowania minikomputera

— odciążenie komputera głównego od szeregu czynności organizacyjnych, jak np. sterowanie przepływem danych, zarządzanie zbiorami, zarządzanie transmisją do zdalnie przyłączonych urządzeń.

Firma TEKNOS opracowała pakiet oprogramowania rezydujący w minikomputerze MICRO 1 i spełniający funkcje logiczne IBM-owskiej metody ISAM oraz niektóre funkcje metody BDAM.

Opracowany system umożliwia dwa sposoby organizacji zbiorów: bezpośredni (*direct*), i indeksowo-sekwencyjny (*indexed sequential*).

Jednostka centralna inicjuje procedury związane z wyszukiwaniem zapisów i utrzymaniem zbiorów za pośrednictwem meldunków o ustalonym formacie.

W przypadku systemu cyfrowego opartego na komputerze, którego system operacyjny nie zawiera procedur współpracy z pamięcią dyskową należy dokonać pewnych zmian w oprogramowaniu systemowym komputera głównego oraz dostosowanie oprogramowania użytkowego do zwiększonych możliwości pamięci zewnętrznej o bezpośrednim dostępie. (MIŃSK 32). Zmiany te nie są niezbędne w przypadku istnienia standardowego oprogramowania i systemu operacyjnego uwzględniającego dyski np. ODRA 1305.

Rys. 2 przedstawia generalny schemat przyłączenia minikomputera do jednostki centralnej i wykorzystania go jako jednostki sterującej.

Podobnie jak i jednostka centralna minikomputer wyposażony jest w szynę we/wy. Jednostka sterująca współpracuje z minikomputerem poprzez elektryczne złącze (dostarczane przez firmę). Kolejnym niezbędnym blokiem jest eletkryczne złącze umożliwiające elektryczne połączenie minikomputera z jednostką centralną. Ten układowy blok jest również opracowany przez firmę dla maszyny MIŃSK 32. Minikomputer traktowany jest przez jednostkę centralną jako urządzenie we/wy i odwrotnie.

W takiej konfiguracji minikomputer, wykorzystując własny multipleksor i adaptery komunikacyjne, może zarządzać transmisją danych stowarzyszoną z systemem cyfrowym.

Jerzy Kurkiewicz
Stanisław Michalak

Konkurenci IBM łączą swe siły

W połowie kwietnia bieżącego roku amerykańskie koncerny CONTROL DATA CORPORATION oraz HONEYWELL Ic. podpisały porozumienie o utworzenie wspólnej filii, która zajmować się będzie pracami badawczo-rozwojowymi w zakresie sprzętu peryferyjnego.

W ten sposób firma CDC kontynuuje proces zacieśniania współpracy z konkurentami „giganta" IBM. Rozpoczął się on już w styczniu 1972 roku podpisaniem umowy o kooperacji naukowej i produkcyjnej między CDC a NATIONAL CASH REGISTER.

W ramach tej umowy utworzono wspólne przedsiębiorstwo o nazwie COMPUTER PERIPHERICALS Inc. (CPI), które posiada obecnie na terenie Stanów Zjednoczonych cztery zakłady produkujące drukarki wierszowe oraz urządzenia wejścia-wyjścia na taśmę dziurkowaną i karty. W grudniu 1973 roku CDC oraz NCR utworzyły ponadto spółkę ASL (ADVANCED SYSTEMS LABORATORY), która prowadzi prace badawczo-rozwojowe nad architekturą jednostek centralnych (jm).

Niezadowolenie użytkowników sprzętu

Dyrekcje 14 zachodnioeuropejskich, amerykańskich i japońskich linii lotniczych, a między innymi ALIATALIA, SWISSAIR, BRITISH AIRWAYS, JAPAN AIRLINES i PAN AMERICAN, wystąpiły wspólnie z ostrą kampanią prasową przeciwko koncernowi IBM. Wszystkie zainteresowane linie lotnicze korzystają od wielu lat ze sprzętu koncernu amerykańskiego oraz uczestniczą w międzynarodowym systemie rezerwacji miejsc (IPARS) który także opiera się na eksploatacji komputerów IBM.

Przedsiębiorstwa lotnicze uskarżają się na to, iż w bieżącym użytkowaniu maszyny seri „370", (głównie modele 158) nie osiągają parametrów eksploatacyjnych gwarantowanych przez firmę, a praktyczny współczynnik awaryjności jest o wiele wyższy niż w przypadku wycofanych maszyn serii „360".

Kierownictwo koncernu IBM nie zajęło jak dotąd publicznego stanowiska w tej ważnej sprawie, twierdząc, iż wszelkie problemy techniczne powinny być rozwiązywane w bezpośrednich kontaktach z użytkownikami.

Skargi linii lotniczych wydają się potwierdzać opinie niektórych fachowców, którzy uważają, że amerykański gigant z przyczyn konkurencyjnych przedwcześnie zdecydował się wyjść na rynek z niedopracowaną do końca serią maszyn. (jm)

Rynek minikomputerów

Specjaliści od markietingu, firma FROST i SULLIVAN opublikowała 536-stronnicowy raport dotyczący rynku minikomputerowego w Zachodniej Europie. Oceniają oni, że obroty na tym rynku stanowiące w 1973 r. równowartość 67 milionów funtów brytyjskich wzrosną siedmiokrotnie do 448 milionów w 1983 r. Całkowity obrót w ciągu najbliższych 10 lat ma wynieść 2.700 milionów, w tym — RFN — 690 milionów, W. Brytania około 590 milionów i Francja — 531 milionów funtów.

Przy tych ocenach zostawiono margines na zwiększające się zastosowanie mikroprocesorów opartych na jednej strukturze układowej (czip), „co stanowi zagrożenie wielu potencjalnych zastosowań minikomputerów". Braki materiałów półprzewodnikowych i wysokie ceny ropy naftowej nie wywrą, zdaniem ekspertów, większego wpływu na ten rynek.

Obecnie na rynku europejskim są oferowane minikomputery około 55 producentów i przewiduje się zwiększenie tej liczby (zwłaszcza producentów japońskich). Raport stwierdza, że wejście na rynek w ciągu kilku lat „było stosunkowo proste i nie wymagające ani wysokich nakładów finansowych, ani też znacznego potencjału w zakresie projektowania i wytwarzania. Mimo to duże firmy były skłonne do ignorowania tej dziedziny i wielu producentów minikomputerów są to małe firmy oparte czasem na dotacjach rządowych. Walka konkurencyjna jest tu szczególnie ostra: „Dla utrzymania się na rynku wszyscy, nawet najwięksi producenci, muszą się specjalizować i być przygotowani na konieczność dostarczania bardziej złożonych wyrobów niż obecnie dostępne".

Ocenia się, że w ciągu najbliższej dekady prawie 33% minikomputerów znajdzie zastosowanie w teledacji, następne 30% w sterowaniu procesami technologicznymi a 20% do celów zarządzania. Dla potrzeb systemów laboratoryjnych znajdzie zastosowanie około 11% minikomputerów. Innymi możliwościami zastosowań, jest dziedzina przygotowania danych oraz sterowanie urządzeń peryferyjnych.

Producenci minikomputerów wydają się być zgodni z głównym wnioskiem raportu — przewidywaniami wzrostu sprzedaży, zapotrzebowania na bardziej kompletne wyroby i zastosowania do celów zarządzania. W zakresie bardziej kompletnych wyrobów, żaden z producentów nie zaprzecza, że nie prowadzi w tym kierunku prac.

Peter Benstead, dyrektor GENERAL AUTOMATION, wskazuje, że podczas gdy do tej pory typowy użytkownik systemu minikomputerowego, głównie pracownik naukowy wyższej uczelni, był zadowolony, gdy mógł modyfikować sprzęt i sam opracowywać oprogramowanie, to obecni użytkownicy są bardziej zainteresowani w instalacjach „pod klucz". Ten punkt widzenia jest podkreślany przez takie firmy jak COMPUTER TECHNOLOGY i DIGICO, które charakteryzując swoje wyroby mówią o „totalnych możliwościach systemu" (*total systems capability*).

Prowadzone są również przygotowania do pokrycia wzrostu zapotrzebowania na minikomputery do celów zarządzania.

Wspomniany Peter Benstead tak podsumował główne wnioski raportu: „Nie tylko, że zgadzamy się z przewidywaniami FROST'a i SULLIVAN'a, lecz musimy je realizować". (lj)

Półprzewodnikowe ekrany telewizyjne

WESTINGHOUSE ELECTRIC Corp (USA) prowadzi badania nad półprzewodnikowymi ekranami telewizyjnymi. Ostatnim osiągnięciem jest ekran o powierzchni 38 cm² zawierający 36.000 elementów. Zasadnicze układy elektroniczne składają się z systemu wybiorczego i tranzystorów regulujących jasność, na które napylona jest warstwa fosforu siarkowo-cynkowo-kadmowego. Zewnętrzną elektrodę stanowi napylona na warstwę fosforową warstwa złota i tlenku ołowiu.

Tranzystorowy ekran eliminuje dwie podstawowe wady istniejących do tej pory płaskich ekranów: każdy element obrazu może zmieniać swoją jaskrawość dla uzyskania cieni i kolorów oraz może być sterowany niezależnie od innych elementów (lj).

Układy COS/MOS

Obecnie firma RCA sprzedaje na rynku europejskim scalone obwody logiczne typu COS/MOS w ilościach stanowiących ponad 50% całkowitego rynku tych elementów. Układy COS/MOS zyskują stale na popularności ze względu na niewielkie wartości prądów zasilania i stosunkowo dobre charakterystyki czasowe. RCA produkuje 96 różnych odmian obwodów COS/MOS, począwszy od prostych inwertorów przez różnego typu rejestry aż do pamięci 256-bitowych RAM (lj).

Bilans firmy BENSON

Według informacji zamieszczonych w numerze 172—173 tygodnika „INTERELECTRONIQUE" z dnia 5 maja br. obroty firmy BENSON wzrosły z 19 milionów franków francuskich w r. 1973 do 23 milionów w 1974 r. Liczba zainstalowanych maszyn infograficznych BENSONA przekroczyła w Europie 1000 sztuk, co stanowi około 80% francuskiego rynku automatycznego. (kr)

Obroty największych firm komputerowych świata

Największy producent komputerów, firma IBM zarejestrowała w r. 1974 obroty w wysokości 12.675 mln dol. zysk za ten okres wyniósł 16,6% 1.837 mln dol. w porównaniu do 15,3% w r. 1973.

Dla porównania inny wielki koncern produkcji sprzętu informatycznego BURROUGHS osiągnął w r. 1974 łączne obroty w wysokości 1532 mln dol. tj, o około 300 mln dol. mniej niż wyniósł za ten sam okres zysk firmy IBM. (tw)

Terminy zawarte w poniższym, piątym z kolei odcinku propozycji terminologicznych oparte zostały na wydanej w grudniu ub. r. normie ISO 2382/VI: Data processing-Vocabulary. Section 06: Preparation and handling of data (First edition 1974-12-15).
Podobnie jak w poprzednich odcinkach, zachowano strukturę i nomenklaturę klasyfikacji tematycznej haseł przyjętą w dokumencie oryginalnym.

Sekcja 06. : Przygotowanie i operowanie danymi *(Preparation and handling of data)*

06.01. Terminy ogólne *(General terms)*

- **czytać** *(to read)* — pobierać dane z urządzenia pamięciowego, nośnika danych lub z innego źródła
- **pisać** *(to write)* — rejestrować dane w sposób trwały lub okresowy w urządzeniu pamięciowym lub na nośniku danych
- **kopiować** *(to copy)* — odczytywać dane źródłowe i nie zmieniając ich treści, zapisywać w innym miejscu w postaci fizycznej, która może być różna od postaci źródłowej. **Przykład:** kopiować karty dziurkowane na taśmie magnetycznej
- **reprodukować** *(to duplicate)* — przepisywać dane z postaci źródłowej na postać wynikową, zachowując ich identyczną postać fizyczną. **Przykład:** Dziurkować karty zgodnie z układem i treścią perforacji wzorcowej

06.02. Wprowadzanie i wyprowadzanie *(Input and output)*

- **dane wejściowe** *(input data, input)* — dane pobierane lub przeznaczone do pobrania przez system komputerowy lub jego część
- **wprowadzanie** *(input process, input)* — proces polegający na pobieraniu danych przez system komputerowy lub jego część
- **wejściowy** *(input)* — określenie urządzenia, procesu lub kanału, realizujących wprowadzanie, a także danych lub stanów związanych z wprowadzaniem
- **wprowadzanie bieżące, wprowadzanie uwarunkowane czasowo, wprowadzanie w czasie rzeczywistym** *(real time input)* — pobieranie przez system komputerowy danych wejściowych w granicach czasu określonych przez wymagania innego systemu
- **dane wyjściowe** *(output data, output)* — dane dostarczane lub przeznaczone do dostarczenia z systemu komputerowego lub jego części
- **wyprowadzanie** *(output data, output)* — proces polegający na dostarczaniu danych z systemu komputerowego lub jego części
- **wyjściowy** *(output)* — określenie urządzenia, procesu lub kanału, realizujących wyprowadzanie, a także danych lub stanów związanych z wyprowadzaniem
- **wyprowadzanie bieżące wyprowadzanie uwarunkowane czasowo, wyprowadzanie w czasie rzeczywistym** *(real time output)* — otrzymywanie z systemu komputerowego danych wyjściowych w granicach czasu, określonych przez wymagania innego systemu
- **wejściowo-wyjściowy** *(input-output)* — określenie urządzenia, procesu lub kanału, realizujących wprowadzanie oraz wyprowadzanie, które nie muszą przebiegać równocześnie

06.03. Przesyłanie i konwersja *(Transfer and conversion)*

- **przesyłać** *(to transfer. to move)* — przekazywać dane z jednego miejsca i odbierać je w innym miejscu
- **przesyłanie blokowe, przesyłanie blokami** *(block transfer)* — proces przesyłania jednego lub większej liczby bloków danych, zainicjowany przez pojedynczą operację
- **przesyłanie zewnętrzne** *(peripheral transfer)* — proces przesyłania danych pomiędzy dwoma jednostkami urządzeń zewnętrznych
- **przekształcać, transformować** *(to transform)* — zmieniać postać danych zgodnie z ustalonymi regułami bez istotnej zmiany znaczenia tych danych
- **tłumaczyć, translować** *(to translate)* — przekształcać jeden język na inny język
- **konwertować** *(to convert)* — zmieniać reprezentację danych z jednej postaci na inną postać bez zmiany treści informacyjnej tych danych. **Przykłady:** konwersja kodu, konwersja postaci analogowej na postać cyfrową
- **transliterować** *(to transliterate)* — konwertować dane znak po znaku
- **kodować, zakodowywać** *(to encode, to code)* — konwertować dane za pomocą kodu w sposób zapewniający odtworzenie tych danych w postaci pierwotnej
- **dekodować** *(to decode)* — konwertować dane celem doprowadzenia ich do postaci poprzedzającej zakodowanie
- **przepisywać** *(to transcribe)* — przesyłać dane z jednego nośnika danych na inny nośnik, konwertując je w zakresie potrzebnym dla przyjęcia ich przez nośnik odbierający
- **koncentrować, zagęszczać** *(to pack)* — zapisywać dane na nośniku pamięci w postaci zwartej, wykorzystując własności danych i nośnika w sposób umożliwiający odzyskanie pierwotnej postaci danych. **Przykład:** wykorzystywanie na nośniku tych miejsc (bitów lub bajtów), które przy normalnym działaniu pozostały by nie wykorzystane
- **dekoncentrować, rozgęszczać** *(to unpack)* — odzyskiwać pierwotną postać danych z postaci skoncentrowanej

Władysław Klepacz

Białowieża, Kołobrzeg i co dalej?

Refleksje uczestnika

W grudniu 1973 roku Zjednoczenie Informatyki zaproponowało swoim programistom z całej Polski spotkanie w Białowieży. Tam powiedziano też, że produkty programowe ZI powinny mieć cechy produktów przemysłowych a nie chałupniczych. Utworzono termin: przemysłowa produkcja oprogramowania (PPO) i zaczęto się nimi posługiwać w różnych rozmowach i pisać w różnych dokumentach.

W maju 1975 roku w Kołobrzegu odbyło się seminarium „Organizacja, metodyka i technika produkcji oprogramowania". Dostarczone uczestnikom seminarium teksty referatów noszą tytuł: Problemy przemysłowej produkcji oprogramowania.

Można więc chyba powiedzieć, że w ciągu półtora roku coś w tym temacie się działo. Spotkanie Kołobrzeskie stanowi okazję do spojrzenia wstecz oraz w najbliższą przyszłość.

Zacznijmy od pewnych analogii, nie bojąc się, że mogą się wydać może komuś zbyt megalomańskie.

W październiku 1968 roku odbyła się w Garmisch znana już w świecie konferencja na temat technik inżynierii softwareowej — a w rok później na ten sam temat w Rzymie. Materiały tych konferencji wymieniane są w odsyłaczach prawie każdego artykułu dotyczącego inżynierii oprogramowania — już choćby dlatego, że od czasu tych konferencji termin „software engineering" wszedł do powszechnego użytku. Dwie rzeczy są bardzo znamienne — odbyły się tylko dwie takie konferencje — obie organizowane przez NATO Science Committee — bo nie należy chyba sądzić, by materiały ewentualnych następnych były niedostępne publicznie, można więc zadać pytanie dlaczego tylko dwie, skoro sprawa wzbudziła tak powszechne zainteresowanie?

Druga sprawa to obserwacje edytorów J.N. Buxtona i B. Randella, którzy we wprowadzeniu do materiałów z konferencji rzymskiej piszą, że chociaż formalny układ i formalne podobieństwo treści obu konferencji są bardzo zbliżone — to przecież były to zupełnie różne konferencje. Na pierwszej usiłowano zidentyfikować, sklasyfikować i przedyskutować problemy — zarówno merytoryczne, jak i organizacyjnego charakteru — które występują przy dużych przedsięwzięciach software'owych.

Natomiast ... „intencją organizatorów rzymskiej konferencji było poświęcić ją bardziej szczegółowemu rozpatrzeniu problemów technicznej natury, a nie problemom natury organizacyjnej, które tak zaważyły nad konferencją w Garmisch.

Jednakże jeszcze raz (i z sukcesem) poświęcono wiele starań, by przyciągnąć grono uczestników z równie szerokiego kręgu.

Konferencja była z pewnych względów niezbyt podobna do poprzedniej. Nie widać było napięcia, wynikającego z świadomości natknięcia się na powszechne problemy (zjawiska należące do kryzysu software'owego — dop J.D.) — tak widoczne w Garmisch. Dominującą cechą konferencji — przynajmniej w opinii edytorów — stał się brak porozumienia (lack of communication) między różnymi sekcjami uczestników. Koniec końców powaga tego niedostatku porozumienia spowodowała, że samo to zjawisko stało się głównym przedmiotem dyskusji — po zdaniu sobie sprawy, że jest ona odbiciem (na tej konferencji) tego, co się dzieje w świecie. I tak jak zdanie sobie sprawy z powagi kryzysu softwaréowego można uznać za najważniejszy wynik spotkania w Garmisch — tak zdanie sobie sprawy z „luki porozumienia" jest — zdaniem edytorów — najważniejszym wynikiem konferencji w Rzymie".

Tyle Buxton i Russel — czy ich słowa mogą być jakoś odniesione do sytuacji, która (z pewnym opóźnieniem) może dawała się obserwować w Białowieży i Kołobrzegu? Otóż wydaje się, że tak. A jeśli tak, to warto chyba podjąć wysiłki, aby skorzystać z doświadczeń pozytywnych i negatywnych innych osób, organizacji i krajów, w których kryzysy informatyczne zaczęły się wcześniej.

O jakich podobieństwach można mówić między konferencjami w Garmisch i w Rzymie oraz w Białowieży i Kołobrzegu — aby maksymalnie skorzystać z analogii nie musząc odwoływać się do różnicy poziomów czy wielkości rozpatrywanych problemów, które w oczywisty sposób czynią te cztery zdarzenia nieporównywalnymi — jeśli nie chce się uczynić porównania śmiesznym?

W Garmisch spotkali się ludzie, którzy wiedzieli, że napotkali problem, z którym poradzić sobie trzeba, bo nierozwiązany — w poważny sposób wpłynie hamująco na rozwój zastosowań maszyn cyfrowych — problem ten został nazwany kryzysem software'owym. W Garmisch nie wiedziano jeszcze jak sobie z nim poradzić. Na następnej konferencji w Rzymie — zaczęto dyskutować metody rozwiązania kryzysowej sytuacji ... i nastąpił nowy kryzys — brak porozumienia.

W Białowieży spotkali się ludzie, którzy stwierdzili, że wiele z metod, jakie stosowane są przy pisaniu programów w środowiskach — które reprezentują — pozostawia wiele do życzenia pod względem efektywności, że są źle ustawione bodźce ekonomiczne i organizacja pracy. Zostało wylansowane hasło: przemysłowa produkcja oprogramowania. Ale było to przede wszystkim hasło, chociaż może więcej, może pewien kierunek poszukiwań.

W Kołobrzegu spotkali się ludzie, z których część mogła już powiedzieć: spróbowaliśmy pracować innymi metodami, nie są one może rewelacyjne, ale na pewno lepsze. Konkretnymi przykładami zilustrowali swoje osiągnięcia.

Reakcje były bardzo różne: obserwowało się np. wyraźne rozbawienie części uczestników faktem, że w jednej z grup wszystkie prace na maszynie wykonuje jeden z członków grupy, a nie każdy programista uruchamia swoje programy. Na pewno więc można mówić o braku porozumienia i to nawet groźniejszym, niż gdyby to był brak porozumienia w sferze intelektu — jest to brak porozumienia w sferze emocjonalnej. Padły też głosy, że konferencja w swej obecnej postaci nie spełnia zadania — potrzebne są obrady w wąskich, specjalistycznych gronach — wymieniano np. konieczność pracy nad standardami. Czy, jeśli na przyszłym spotkaniu wprowadzimy pracę w sekcjach — nie dojdziemy do podobnego wniosku jak organizatorzy konferencji w Rzymie, że dominującym osiągnięciem naszego kolejnego spotkania było stwierdzenie, że naszą grupę charakteryzuje brak porozumienia?

Ponieważ brak porozumienia jest zdecydowanie negatywną cechą, zastanówmy się, co warto byłoby zmienić w organizacji i w duchu przyszłych spotkań typu białowieskiego i kołobrzeskiego, by naszą refleksją mogło być po roku, że wzajemne porozumienie i wzajemne zrozumienie potrzeb uczestników tych konferencji zaczyna na nich dominować i jest głównym uzasadnieniem celowości tych spotkań.

Następujące czynniki powinny kierować organizatorami i uczestnikami tych konferencji — takie jest przynajmniej nasze odczucie:

- referenci powinni przyjeżdżać tam z nastawieniem, że ich zadaniem jest nauczyć innych metod, które uznali za dobre, a nie chwalić się, że udało im się wynaleźć i/lub zastosować dobre, nowoczesne metody
- uczestnicy konferencji powinni się rekrutować przede wszystkim z grona takich osób, którym przyjemność sprawia stosowanie w pracy coraz to wydajniejszych metod, którzy lubią pracować rozmaicie, tacy, którzy po cichu wyznają zasadę, że lenistwo jest matką wynalazków i dlatego stale coś nowego wynajdują lub przynajmniej stosują
- organizatorzy powinni zapewnić warunki, aby osoby, które przyjadą i nie zechcą, czy nie potrafią trafić do żadnej z powyższych dwóch grup — mogły uznać czas konferencji za spędzony pożytecznie, albo przynajmniej przyjemnie. Może więc warto zorganizować obrady w sekcjach,

może urządzić konkurs na programy rozrywkowe *), może nawet zająć się tworzeniem tradycyjnych standardów i dyskutować jaka powinna być forma znaków graficznych na schematach czynnościowych programów.

A mówiąc bardziej serio i bardziej konkretnie, naszą propozycję uszczegółowiamy następująco.

Można założyć, że przez lata pracy projektanci systemów i programiści nagromadzili sporo doświadczeń programowych. Doświadczenia te jednakże funkcjonują prawie wyłącznie na poziomie jednostkowym lub w niewielkich grupach.

Dla rozwoju metod pracy nad programowaniem w całej sieci Zjednoczenia Informatyki nie mają więc niestety większego znaczenia.

Wykorzystywanie doświadczeń wyróżniających się programistów i projektantów, którzy stosują coraz to nowe, efektywniejsze metody pracy ma dwa aspekty: szkoleniowy i integracyjny.

Daje o sobie znać potrzeba doskonalenia zawodowego zaawansowanych programistów. Takiego szkolenia nie są w stanie zorganizować — zgodnie z potrzebami poszczególnych Ośrodków — organizatorzy zewnętrzni w stosunku do ZETO, ani krajowi producenci maszyn. Ten typ szkolenia należy więc oprzeć na wyróżniających się, doświadczonych programistach i projektantach. Powinni oni być wykorzystywani do prowadzenia krótkich staży dla innych programistów — przekazując im odpowiednio uogólnione własne doświadczenia.

Dodatkową motywacją do podejmowania trudu szkolenia innych powinno być zapewnienie im przez ZI odbycia szkolenia wyższego III stopnia — w interesującym ich przedmiocie za granicą, z obowiązkiem przeszkolenia po powrocie co najmniej kilku programistów z kilku ośrodków ZI. Szkolenie takie trzeba zacząć już teraz.

Zakupy nowoczesnego sprzętu muszą być poprzedzane intensywnym szkoleniem wyższego stopnia. W przeciwnym przypadku sprzęt nigdy nie będzie wykorzystany zgodnie z jego możliwościami.

Oparcie się w przeprowadzaniu szkolenia dla projektantów i programistów ze wszystkich ośrodków na własnej kadrze — na jej najbardziej wyróżniającej się części — będzie miało walor integracyjny — przez przyswajanie i rozpowszechnienie nowych metod pracy i wykorzystywanie doświadczeń gromadzonych przez lata w różnych Ośrodkach.

Stawiamy hipotezę, że warunkiem skutecznego wykorzystania doświadczeń będzie:

— zmotywowanie doświadczonych programistów do uogólniania własnych doświadczeń i przekazywania ich kolegom zainteresowanym podobnymi problemami

— zorganizowanie kontaktów między programistami z różnych ośrodków dla wymiany doświadczeń i odblokowanie psychologicznych przeszkód w tej wymianie

— zachęcenie pracowników do stosowania (a przynajmniej prób stosowania) nowych metod pracy, by doświadczenia były rozwijane przez kolejnych projektantów i programistów

— zorganizowanie zakupów sprzętu najwyższej światowej klasy, by najbardziej doświadczona kadra mogła pracować na najbardziej nowoczesnym sprzęcie — przynajmniej w okresie owych wewnętrznych staży.

Elementem niezbędnym w samokształceniu drogą zdobywania coraz bogatszych doświadczeń jest umiejętność formułowania uogólnień w oparciu o gromadzone doświadczenia. Zdolność ta jest bardzo różna u różnych ludzi i należy ją rozwijać. Zjednoczenie Informatyki nie akcentowało dotychczas konieczności rozwijania tej cechy.

W wyłanianiu spośród programistów kadry instruktorów — taka umiejętność krytycznej oceny projektów i programistów oraz umiejętność formułowania wniosków metodycznych na użytek szerszej rzeszy programistów powinny być w szczególnej cenie.

Konferencje i seminaria do chwili obecnej są organizowane w sposób, który nie zapewnia niestety maksymalnego wykorzystania tego typu doświadczeń.

Pobieżna prezentacja materiałów przez kilku referentów skupiała dyskusję raczej na sprawach drugorzędnych. Sądząc po reakcjach w czasie dyskusji — raczej nie znajdą się tacy, którzy prezentowane metody przeniosą na własny teren lub choćby zrelatywizują oceny własnej metody.

Warto z tego względu inaczej formułować cele takich spotkań i zagwarantować ich realizację.

Celem powinno być przedstawienie przez kilku wybranych programistów własnych lub adaptowanych i sprawdzonych przez ich praktykę — nowoczesnych metod pracy. Takie referaty były już prezentowane w Kołobrzegu. Należałoby jednak takie materiały prezentować w taki sposób, by werbować jak najwięcej chętnych do odbycia przeszkolenia w tym zakresie jeszcze w czasie trwania seminarium.

Aby formę prezentacji materiału i akwizycji uczestników przeszkolenia maksymalnie uatrakcyjnić można byłoby sobie wyobrazić, że seminarium rozpoczyna się serią dyskusji panelowych, w których potencjalni wykładowcy starają się pozyskać maksymalną liczbę słuchaczy na „swój" kurs organizowany w ramach tego samego choćby seminarium polemizując z innymi wykładowcami — pokazując walory „swojej" metody w fachowej dyskusji.

Słuchacze mieliby w ten sposób okazję poznania przedmiotu i poznania człowieka, który w ciągu najbliższych godzin ma ich czegoś nauczyć.

Jeśli zauważymy, że Kołobrzeg ma stać się w najbliższym czasie ośrodkiem targów software'owych i będzie dysponował dobrze wyposażonym ośrodkiem obliczeniowym, szanse rozwinięcia takiej formy „workshop'ów" stają się duże.

W metodach prezentacji referatów i metodach szkolenia panuje u nas szczególnie duża sztampa i rutyna. Widać to jaskrawo, gdy uczestniczy się w coraz częstszych w Polsce wystąpieniach specjalistów z naszej dziedziny z innych krajów. Najwyższy czas zacząć zmieniać sytuację w tej dziedzinie — potrzebne są do tego przede wszystkim pomysły, chęć i wytrwałość.

Białowieża i Kołobrzeg zapoczątkowały nowe procesy w Zjednoczeniu Informatyki — ich kontynuacja i rozwijanie jest niezbędne.

Jerzy Dańda
Irena Malerczyk-Dańda

*) jak się wydaje, większość programów rozrywkowych w naszym kraju należy do jednej klasy: komputerowego aktu kobiecego wynikającego z realizmu określonego możliwościami drukarek wierszowych. Bogate możliwości zagadek i rozrywek innego typu ilustrują odpowiednie działy np. w piśmie SOFTWARE PRACTICE and EXPERIENCE, gdzie np. pojawił się algorytm rysowania dywanu Sierpińskiego.

Seminarium: Organizacja, metodyka i technika produkcji oprogramowania
Kołobrzeg 14-17.V.1975

Organizatorem seminarium był ZETO Szczecin i TNOiK oddział w Szczecinie przy współudziale Klubu Programistów i Zjednoczenia Informatyki.

Obrady konferencji otworzył doc. dr hab. inż. Jan Goliński referatem: „Problemy software'u i zadania jakie w związku z jego produkcją stoją przed Zjednoczeniem Informatyki" [1]).

Głównym kierunkiem rozwoju Zjednoczenia będą systemy użytkowe.

Przewiduje się, że wroku 1980 w zespołach wytwarzających oprogramowanie użytkowe będzie pracowało około 2100 osób, to jest 70% ogółu zatrudnionych w produkcji oprogramowania.

Standardowe wymogi, jakie stawia ZI przed grupą wyrobów to:

— krótki czasokres prac nad systemem (łącznie z jego wdrożeniem), nie przekraczający 6 miesięcy

— możliwość wprowadzania zmian organizacyjnych u użytkownika i jednocześnie w eksploatowanym systemie

— łatwość prowadzenia prac wdrożeniowych, bez potrzeby angażowania własnych wysokokwalifikowanych kadr.

Kolejnym krokiem, jaki czeka zakłady Zjednoczenia to stopniowe przejście od dotychczas stosowanych pakietów użytkowych do powtarzalnego oprogramowania parametryzowanego.

Przewidziane jest również wykorzystywanie do produkcji software'u generatorów oprogramowania. Komputer w ramach tego

[1]) materiał publikowany w nr 7—8/75 Informatyki.

procesu będzie elementem znacznie wspomagającym proces projektowania i programowania.
Zjednoczenie zaangażuje około 30% swego software'owego potencjału wytwórczego w projektowanie i oprogramowanie systemów powtarzalnych. Zespół realizujący to zadanie będzie w r. 1980 liczył około 900 osób. Przewidywane zasoby oprogramowania powtarzalnego pozwolą w latach 1977—1980 stosować to oprogramowanie w systemach obiektowych.
Innym kierunkiem działania ZI jest praca nad softwarem maszyn Jednolitego Systemu. Maszyny JS staną się w przyszłości bazą dla Zjednoczenia.
Prace nad oprogramowaniem prowadzone są w OBRI w ramach tematu ASU jak również przez Zakład R-POL-SYSTEM.
Drugi referat na temat polityki sprzętowej ZI wygłoszony został przez z-cę dyr. d/s technicznych mgr inż. Ryszarda Terebusa.
Najpilniejszym zadaniem stojącym obecnie przed ZI to nie rozwój bazy sprzętowej ośrodków ZETO (w roku 1974 zakupionych zostało 16 komputerów), lecz jakościowy. Bolączką wszystkich zakładów ZETO jest ich niedozbrojenie. Przyczyną tego stanu rzeczy jest to, że producent „maszyn MERA-ELWRO dostarcza je w standardowej konfiguracji niedostosowanej do potrzeb ZETO. Brak jest przede wszystkim dodatkowych jednostek pamięci taśmowych, drukarek czytników kart oraz pięty achillesowej całego systemu w przypadku ODRY 1305 operatorskiej FACIT. Takie wyposażenie uniemożliwia wprowadzenia na codzień wieloprogramowości a więc właściwego wykorzystania mocy obliczeniowej. Jedynie dwa zakłady ZETO Wrocław i Gdynia wyposażone są w jednostki dysków i stosują system operacyjny GEORGE 3.
W związku z coraz dotkliwszym brakiem papieru na rynkach światowych obserwuje się tendencję odchodzenia od nośników papierowych i nanoszenia danych bezpośrednio z klawiatury na taśmę magnetyczną lub czytniki dokumentów.
Dlatego też w najbliższych latach poszczególne ZETO będą wyposażone w taki sprzęt.
Przewiduje się też rozwój systemów abonenckich zarówno w oparciu o ODRĘ jak i o RIADA.
W godzinach popołudniowych odbyło się spotkanie z doc. dr Markiem Greniewskim, który omawiał z uczestnikami nowe kierunki inżynierii softwarowej.
W drugim dniu obrad z-ca dyr. mgr inż. R. Terebus przedstawił referat wprowadzający pt. „Problemy organizacji targów software'owych".
Wychodząc z założenia, że wymiana software'u jest złożonym przedsięwzięciem nie tylko organizacyjnym ZI postanowiło zorganizować w końcu 1975 r. I Targi Softwarowe ZI i kontynuować tę imprezę corocznie. Proponuje się aby stałym miejscem targów było miasto Kołobrzeg.
Celem targów jest:
— prezentacja dorobku poszczególnych przedsiębiorstw ZETO
— wymiana oraz obrót oprogramowaniem pomiędzy krajowymi ośrodkami i organizacjami
— podniesienie poziomu merytorycznego oraz technik dokumentowania programów oraz postawienie odpowiednich wymagań wystawcom.
— zaproszenie zagranicznych firm software'owych dla rozeznania problemu nie tylko importu oprogramowania, ale również badanie możliwości eksportu.
Powołana została rada programowa d/s targów której zadaniem jest przystosowanie targów pod względem merytorycznym.
Kolejnym elementem drugiego dnia seminarium była prezentacja osiągnięć Klubu Programistów przy ZI. Sesji tej przewodniczył kol. Roman Jeliński z ZETO Gdynia.
W godzinach popołudniowych odbyło się spotkanie dyrekcji ZI z uczestnikami seminarium.
Kolejny III dzień obrad poświęcony był organizacji, metodyce i technice produkcji oprogramowania.
Temu problemowi poświęcona jest większość referatów zawartych w dostarczonych uczestnikom materiałach dlatego nie będzie on tu szczegółowo omawiany.
Dyskusja i wypowiedzi uczestników koncentrowały się wokół następnych referatów:
1. Psycho-społeczne i organizacyjne problemy gospodarki kadrowej ZI
2. Aspekty metodyczne i organizacyjne produkcji oprogramowania
3. Organizacja prac nad oprogramowaniem systemów przetwarzania danych na zasadach pracy badawczej
4. Zastosowanie COBOL-u w technikach programowania modularnego.

Seminarium potwierdziło konieczność częstszego organizowania tego typu imprez lecz o charakterze bardziej specjalistycznym z mniejszą liczbą uczestników.

Jerzy Kurkiewicz

Założenia organizacyjno-programowe Targów Software'owych Zjednoczenia Informatyki „ZETO"

Wychodząc z założenia, że wymiana software'u jest złożonym przedsięwzięciem nie tylko organizacyjnym, ZI postanawia zorganizować I-sze Targi Software'owe ZI i kontynuować w/w imprezę corocznie. Proponuje się aby stałym miejscem targów był Kołobrzeg.

CELEM TARGÓW JEST:
— prezentacja dorobku poszczególnych przedsiębiorstw ZETO,
— wymiana oraz obrót software'm pomiędzy krajowymi ośrodkami i organizacjami
— podniesienie poziomu merytorycznego oraz techniki dokumentowania programów przez postawienie odpowiednich wymagań wystawcom
— zapraszanie zagranicznych firm software'owych dla rozeznania problemu nie tylko importu oprogramowania, ale również badanie możliwości eksportu.

FORMA TARGÓW

Przewiduje się stoiska poszczególnych wystawców w formie dokumentacji wdrożeniowej.

WARUNKI TRANSAKCJI

Dla każdego programu wystawca powinien zabezpieczyć:
— doradztwo
— konsultacje
— sprzedaż programu
— wdrożenia
— serwis.

Przedmiotem transakcji może być kompleks w/w zagadnień lub jeden albo kilka z nich.

RODZAJE PROGRAMÓW
— oprogramowanie narzędziowe (ten dział software'u zdecydowanie należy preferować),
— programy lub pakiety dotyczące zarządzania ośrodkami obliczeniowymi,
— pakiety użytkowe, które mogą być powielane dla określonej dziedziny gospodarki narodowej,
— inne opracowania, które zostaną zakwalifikowane przez Radę Programową Targów powołaną przez ZI,
— projekty ośrodków obliczeniowych, systemów cyfrowych, mebli specjalnych, projektów organizacyjnych itp.

UWAGA: Programy mogą być zorientowane na komputery JS; ODRA-1300 i maszyn im kompatibilnych oraz minikomputery MERA 300 i MERA 400.
W wyjątkowych wypadkach mogą dotyczyć innych komputerów pod warunkiem ich łatwej adaptacji na w/w systemy.

ORGANIZATOR: ZI i przedsiębiorstwa, Klub Programistów oraz inne uzgodnione z ZI organizacje.

IMPREZY TOWARZYSZĄCE:

W okresie trwania targów przewiduje się imprezy towarzyszące jak np.:
— dzień obliczeń inżynierskich
— seminaria
— odczyty
— dyskusje panelowe itp.

ZADANIA ZI:

— powołanie stałej Rady Programowej Targów i Komitetu Organizacyjnego,
— koordynacja reklamy i wydawnictw,
— przygotowanie druków standardowych umów i porozumień,
— powołanie w Kołobrzegu stałej komórki organizacyjnej d/s targów,
— zapraszanie firm zagranicznych, central handlowych oraz osób i organizacji.

Szczegółowe warunki uczestnictwa:
— wymagania merytoryczne odnośnie przedmiotów targowych proponuje się zlecić OBRI,
— warunki organizacyjne ustalą odpowiednie służby w ZI.

I moduł systemu „Gospodarka materiałowa", „Ewidencja obrotów i stanów magazynowych"

Opracowany w Resortowym Ośrodku Informatyki Przemysłu Chemicznego „ETOCHEM" zakładowy system EPD „Gospodarka materiałowa („GM") składa się z następujących modułów:

I. Ewidencja obrotów i stanów magazynowych.

II. Gospodarka materiałowa w sferze zaopatrzenia.

III. Gospodarka materiałowa w sferze produkcji z uwzględnieniem zagadnienia materiałowego zabezpieczenia produkcji.

System opracowano dla przedsiębiorstw przemysłu chemicznego.

Przedsiębiorstwa, na których opierano się podczas prac nad systemem, charakteryzują się zróżnicowanym asortymentem produkcji, a tym samym i pozycji asortymentowych zaopatrzenia, zatem musi się on oznaczać dużą elastycznością.

Oprogramowanie Systemu, głównie w języku PLAN i częściowo COBOL, determinuje jego eksploatację na maszynach ODRA serii 1300 oraz ICL serii 1900. Częstotliwość przetwarzania zależy od potrzeb użytkownika, najczęściej jednak stosuje się przetwarzanie miesięczne.

„Ewidencja obrotów i stanów magazynowych" jest podstawowym modułem całego systemu. Prawidłowe wdrożenie i dalsza eksploatacja tego modułu stanowią podstawowe warunki uzyskania wszystkich efektów wprowadzenia pełnego systemu „GM" w przedsiębiorstwie.

Moduł ten ma na celu ujęcie i uporządkowanie wszelkich danych źródłowych dotyczących ewidencji magazynowej, a w szczególności:

— stworzenie i prowadzenie bazy indeksowej dla wszystkich materiałów
— ewidencję obrotów i stanów magazynowych
— rozliczenie zużycia materiałów
— rozliczenie inwentaryzacji
— rozliczenie przecen materiałów
— dostarczenie informacji do sprawozdawczości zewnętrznej.

W celu zapewnienia uniwersalności tego modułu zaprojektowano go a następnie oprogramowano wykorzystując szeroki zakres informacji dotyczących ewidencji magazynowej, przy czym programy wczytujące dane źródłowe dopuszczają w zasadzie dowolne wzory dokumentów źródłowych. Jak wiadomo w przedsiębiorstwach wykorzystywane są różne wzory dokumentów, a także obowiązują różne sposoby ich wypełniania. Ponieważ sprawa generalnego ujednolicenia dokumentacji źródłowej jest zadaniem przyszłościowym, głównie ze względu na indywidualizowanie tych problemów u różnych użytkowników, opracowano rozwiązanie pozwalające na wczytanie danych zapisanych w dowolnym układzie na maszynowym nośniku informacji. Jedynym ograniczeniem jest: występowanie pewnego zakresu informacji niezbędnej dla prawidłowego działania systemu i konieczność symbolizacji dokumentów źródłowych dwucyfrowym kodem jednakowym dla każdego typu dokumentów. Kod ten może być różny u każdego z użytkowników.

Identyfikacja informacji z dokumentów następuje na podstawie parametrów określających kolejność ich występowania na maszynowych nośnikach informacji. Parametry te ustalone są dla każdego użytkownika i są zarazem parametrami dla programu wczytującego dane wejściowe.

Informacje z dokumentów źródłowych mogą być przenoszone zarówno na karty perforowane jak i na taśmę 8-mio kanałową.

W celu zapewnienia wewnętrznej zgodności modułu każdemu wczytanemu dokumentowi zostaje nadany dwucyfrowy symbol już jednakowy dla dokumentów tego samego typu bez względu na ich pierwotną postać. Zaprezentowane tu rozwiązanie pozwala na wdrożenie „Ewidencji obrotów i stanów magazynowych" dla szerokiego grona użytkowników bez napewno kłopotliwej konieczności zmiany dokumentacji źródłowej oraz sposobu wypełniania dokumentów. Oczywiście konieczne jest uporządkowanie tej dokumentacji i ujednolicenie wewnątrz jednego zakładu.

W module „Ewidencja obrotów i stanów magazynowych" podstawowe znaczenie mają dwa zbiory główne: kartoteka indeksowa oraz kartoteka magazynowa. Kartoteka indeksowa jest podstawowym zbiorem dla całego systemu „Gospodarki Materiałowej" i zawiera podstawowe informacje o każdym materiale takie jak: nazwa, indeks materiału, cena ewidencyjna, jednostka miary itd. Kartoteka ta może zawierać dwa indeksy materiałowe. Jeden, stosowany wewnętrznie przez użytkownika tzn. stosowany na dokumentach magazynowych i drugi zgodny z symboliką SWW, który występuje na tabulogramie wynikowym. Oba te indeksy można powiązać jak np. ma to miejsce w Fabryce Kosmetyków POLLENA-URODA, gdzie indeks zgodny z symboliką SWW zawiera 16 znaków, a w tym ostatnie 7 stanowi symbol wewnątrzzakładowy.

Kartoteka magazynowa zawiera aktualne stany materiałów w magazynach a także narastające wartości przychodów i rozchodów z podziałem na ich rodzaje. Kartoteka ta aktualizowana jest poprzez dokumenty magazynowe dostarczane do ośrodka obliczeniowego.

Należy tutaj wspomnieć, że każdy z użytkowników może indywidualnie oznaczyć dokumenty dotyczące zakupu, sprzedaży czy zużycia: informacje te wykorzystywane są w formie parametrów do programu aktualizującego kartotekę magazynową.

Moduł — „Ewidencja obrotów i stanów magazynowych" daje możliwość uzyskania szerokiego zakresu tabulogramów wynikowych. Ponieważ większość z nich jest uzyskiwana niezależnie od siebie (równolegle), każdy z użytkowników może żądać tabulogramów zgodnie z jego potrzebami.

Tabulogramy wynikowe można podzielić na następujące grupy:

— dotyczące stanów i obrotów magazynowych
— dotyczące rozliczenia inwentaryzacji i przeceny
— zestawienia obrotów miesięcznych w układzie gałęziowym
— zestawienia poszczególnych rodzajów dokumentów.

Omówiony moduł został wdrożony w 1974 r. w kilku przedsiębiorstwach przemysłu chemicznego: w lutym 1975 r. po wprowadzeniu dodatkowych usprawnień w oprogramowaniu spełnia swoje zadanie jeszcze lepiej (skrócenie czasu pracy maszyny).

Z uwagi na wzrastające znaczenie gospodarki materiałowo-towarowej przewiduje się dalsze usprawnienia a niezależnie i poszerzenia tematyczne w całości opracowania systemu „GM".

Witold Jędrzejewski

Komputery w muzyce (4)

Po omówieniu teorii i praktyki różnych muzycznych zastosowań komputerów, które prezentowaliśmy w ostatnich trzech numerach, pora na przegląd osiągnięć dokonanych do tej pory eksperymentów w zakresie kompozycji komputerowej. Jest to bez wątpienia najtrudniejsze, ale i najciekawsze spośród muzycznych wcieleń maszyn cyfrowych.

Jedną z pierwszych prac nad kompozycją komputerową wykonali w latach pięćdziesiątych L. Hiller i L. Lscacson na Uniwersytecie w Illinois. Była to kompozycja na kwartet smyczkowy zatytułowana „Illiac Suite" jako, że użyto tu maszyny cyfrowej Illiac. Kontynuacją podjętego kierunku stała się nagrana w 1963 roku „Kantata komputerowa". Muzykę tę wydano na płycie „Computer Music from the University of Illinois" (Heliodor H 25033) a prawa nią rządzące zostały omówione w licznych artykułach i w książce „Expermiental Music". Procedura, której użyto w „Illiac Suite", była dość złożona — wzorowała się na programach analizy związków molekularnych (Hiller jest z zawodu chemikiem). Opierała się na procesach Markowa, czyli statystycznych modelach procesów o zmiennej losowej zależnej. Prawdopodobieństwa osiągania przez zmienną losową odpowiednich przyszłych wartości zależą w tych procesach jedynie od jej aktualnej wartości, nie zależą natomiast od wszystkich poprzednich, które do tej pory zmienna losowa przyjmowała. Jeśli za zmienną losową przyjąć brzmienie kolejnych dźwięków, to prawdopodobieństwa usłyszenia dźwięków z jakiegoś zbioru zależeć będą tylko od tego, jaki dźwięk je poprzedzał. Wprowadzenie procesów Markowa nie dawało sekwencji zupełnie przypadkowych, ale znacznie rozluźniało reguły odtwarzanych utworów. Według opinii krytyków przetworzone w ten sposób fragmenty muzyki klasycznej, odbierało się raczej jak utwory Bartoka". „Computer Cantata" została stworzona przy użyciu systemu MUSICOMP opracowanego przez L. Hillera i R.A. Backera. Zawiera on zbiory rozkazów narzucających losowe parametry wytwarzanym dźwiękom. Użytkownik maszyny może dodać do nich własne instrukcje programowe wprowadzając do powstającej melodii swój oryginalny styl. W systemie tym występują także zbiory rozkazów porządkujących uzyskane wartości losowe. Przykładem takiego zbioru jest podprogram MATCH, korelujący liczby z dwóch ciągów. Jeśli ciągi te przedstawimy jako dwie równoległe proste, na których odkłada się odcinki o długości proporcjonalnej do wartości liczb odpowiadającego im ciągu, to podprogram MATCH tak przedstawia te odcinki, aby otrzymać najwięcej połączeń odcinków leżących na obu prostych w tej samej długości od punktu początkowego. Jest to więc zadanie maksymalizacji równości sum częściowych dwu ciągów liczbowych. Jeśli liczby z tych ciągów (czyli odcinki na prostych) będą oznaczały czasy trwania kolejnych dźwięków na dwóch instrumentach, to operacja taka zmaksymalizuje ilość momentów, w których te instrumenty jednocześnie przystąpią do wykonywania nowego dźwięku.

L. Hiller

Metody opracowane w Illinois znalazły wielu naśłowców i były dalej rozwijane. Wspominamy już w poprzednich artykułach Szkot O'Beirne zastosował je nawet do utworów granych na kobzie.

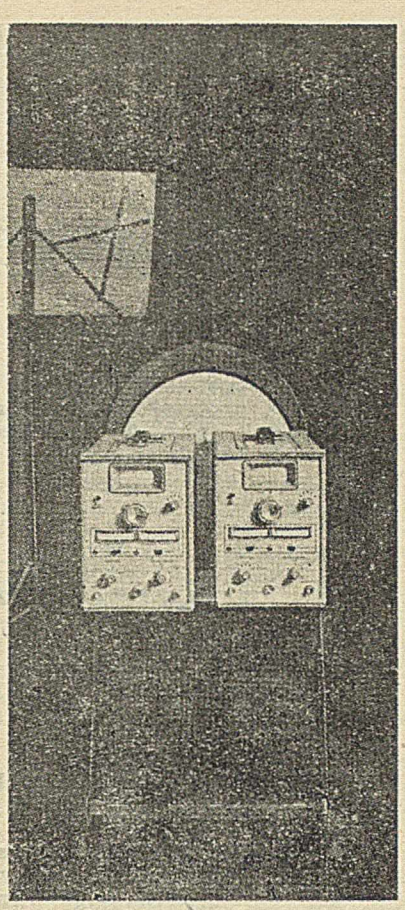
Iannis Xenakis, Grek z pochodzenia, przyjął dla wyboru parametrów swoich kompozycji reguły wychodzące z podstaw statystyki matematycznej. Przebywając w Paryżu współpracował on z francuskim oddziałem firmy IBM i osiągnął pierwsze pozytywne rezultaty w 1962 roku. Program w języku FORTRAN pozwalał na komponowanie różnych melodii przeznaczonych na kwartet smyczkowy oraz na cztery do dziesięciu instrumentów. Utwory te zarejestrowano na płycie „The Music of Iannis Xenakis" (HMV ASD 2442). Kwartet smyczkowy wykonywał także kompozycje holenderskiego matematyka Lamberta Meertensa, który oparł swoją koncepcję na algorytmach służących do automatycznego nauczania. Martens określał stany dozwolone w utworze, a maszyna każdorazowo sprawdzała, czy kolejny krok mieści się w obszarze dopuszczalnym. Jeśli nie, wówczas proponowała inne wartości losowe.

Prace Xenakisa podjęli kompozytorzy francuscy Pierre Barbaud i Marcel Goldmann. Szczególnie interesujące są utwory Barbauda „French Gagaku" i „Machinamentum Primineuse" nagrane na płytach załączonych do przeglądu informacyjnego o systemach produkowanych przez Heneywell Bull.

Barbaud związał się z tą firmą w 1958 roku, gdy wygrał konkurs na najlepszy program w języku FORTRAN dla tworzenia kompozycji muzycznych przez maszynę H 516 produkcji Honeywell Bull. I do muzyki zresztą trafił Barbaud przypadkowo — dzięki temu, że zatrudniony był w dziale muzyki Biblioteki Narodowej — choć studiwoł filologię i języki starożytne.

Inny system kompozycyjny nazwany ZASP zaproponował Alan Sutcliffe. W tym systemie człowiek ustala ogólne założenia dla utworu, resztę zaś określają serie liczb losowych generowane przez program główny. Nawet przy komponowaniu bardzo krótkich utworów odwołanie się do programu głównego następuje ponad t siąc razy. Pierwszą liczbą losową jest wartość z przedziału między 200 a 600, ustalająca ile sekund ma trwać utwór. Następne określają czas trwania poszczególnych części przy założonej długości całego utworu (np. dla 540 sekundowego utworu czasy trwania części mają ograniczenie od 50 do 100 sekund i mogą wynosić: 90, 100, 85, 95, 100, wybierane są bowiem zwykle czasy bliskie górnej granicy).

Podobnie każda z części jest dzielona na fragmenty. W dalszym kroku uzyskuje się siedem zbiorów wartości precyzujących konkretne tony w każdym z fragmentów. Charakterystyczne dla ZASP-u jest to. że do podziału na tych wszystkich poziomach wykorzystuje on ten sam rodzaj działania, a zatem ten sam program, co z punktu widzenia komputera jest niezwykle oszczędne.



muzyki. S. W. Smolinar napisał z myślą o tym assembler EUTERPE, dzięki któremu muzyk określa parametry przekazywanego komputerowi utworu w zapisie tradycyjnym, a maszyna sama przetwarza je na własne kody liczbowe. Smolinar zwrócił ponadto uwagę na podobieństwa między formalnymi regułami kompozycji a zasadami pisania programów oraz na zbieżności w procesach wykonywania muzyki i programów. Uchwycenie tych analogii sprzyja zachowaniu jedności ideowej muzyki i narzędzia, które służy do jej komponowania.

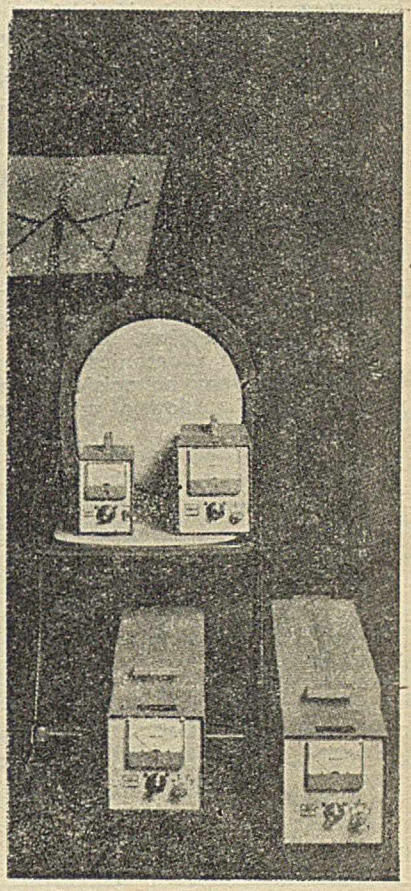
Nowości techniczne w świecie komputerów, zwłaszcza zaś miniaturyzacja układów i powstawanie małych maszyn, również nie pozostały przez kompozytorów nie zauważone. W 1972 roku P.H. Knowlton opublikował wyniki użycia minikomputera PDP w systemie muzycznym. System nia się łatwością wprowadzania danych i zapisywania rezultatów. Na wejściu maszyny znajduje się czytnik kart z kodowanymi parametrami muzycznymi, ale parametry te można także wprowadzać poprzez klawiaturę organów elektronicznych przyłączonych również do komputera. Na wyjściu znajduje się monitor ekranowy wyświetlający zapis kompozycji, pisak rejestrujący go na papierze i organy interpretujące ten zapis. Można więc jednocześnie widzieć i słyszeć wszystko, co wprowadzamy

Wyniki pracy maszyny podawane są w postaci drukowanej partytury i taśmy dziurkowanej określającej parametry każdej z nut.

Mogą one być odtworzone przez towarzyszący komputerowi sprzęt lub przekazane do studia muzyki elektronicznej dla dokonania nagrań. Ze względu na zastosowanie instrumentów elektronicznych Sutcliffe użył zapisu operującego skalą ok. 90 nut w oktawie.

Ponadto nuta opisana jest czasem narastania, ustalania się i opadania sygnału oraz odcinkiem ciszy przed następną nutą. Dane na temat nuty wyrażane są przez siedem liczb (natężenie i postać fali dźwiękowej jest również przedstawione numerycznie). Przez wprowadzenie ograniczonej losowości Sutcliffe otrzymał więc muzykę o interesującej zmienności strukturalnej. Uniknął przy tym monotonii, jaką niesie z sobą przypadek nieukierunkowany, dający podczas słuchania z dużej odległości, efekty statystycznej średniej — jednostajnego szumu.

Wraz z rozwojem teorii programowania punkt ciężkości poszukiwań kompozytorskich przesunął się z samych maszyn na budowę ich programów. Powstawanie języków problemowo zorientowanych tworzonych dla szczególnych zastosowań komputerów, spowodowało, że wielu badaczy postawiło sobie za cel opracowanie języka programowania dla potrzeb

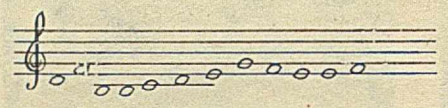


do maszyny i to, co z niej uzyskujemy.

Bardzo ciekawe rozwiązanie zaproponowali dwaj weterani muzyki komputerowej L. Hiller i J. Gage. Oparli je na algorytmach zaczerpniętych z heksametrycznych przepowiedni chińskiej księgi wróżb „I Ching" liczącej ponad 4 tysiące lat. Zestaw wykonawczy obejmuje od jednego do pięćdziesięciu jeden magnetofonów i od jednego do siedmiu klawesynów. Stąd też wziął się kryptonim system — HPSHD — skrót słowa harpsichord (klawesyn). Nagrania HPSCHD dla pięćdziesięciu jeden magnetofonów i trzech klawesynów dokonano na płycie Noesuch HJ 1224. Wszystkie magnetofony odwtarzają jednocześnie muzykę komputerową, natomiast na klawesynach grane są utwory znanych kompozytorów ubiegłych stuleci.

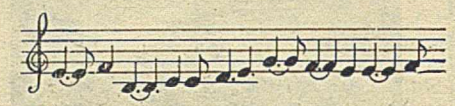
Przy koncertowych wykonaniach HPSCHD publiczność może swobodnie poruszać się między instrumentami. Słyszy się wówczas różne dźwięki, zależnie od wybranej drogi. Uzyskanie najwierniejszego przekazu w nagraniu płytowym — wrażeń odbieranych przez poruszającego się słuchacza — oto zadanie, do którego również zaangażowano komputer. Symuluje on ruch przez zmiany sterowania kopii zapisu w pięciosekundowych interwałach. Samo zaś wykonanie HPSCHD nie miało ograniczeń czasowych i mogło trwać kilka godzin dostarczając bez przerwy nowych porcji muzyki urozmaiconej dodatkowo przez grę świateł 64 rzutników oświetlających salę abstrakcyjnymi zestawami kolorów.

Muzyka komputerowa weszła w etap, na którym opanowana została technika tworzenia nowych dźwięków i budowania z nich kompozycji, wciąż brak jest jednak automatycznych metod ocen tych wyników. Sprzężenie zwrotne ocen z procesem kompozycyjnym selekcjonowałoby powstający materiał muzyczny, wybierając z całej maszynowej produkcji tylko utwory wartościowe lub nowatorskie. Nie istnienie takich metod zmusza kompozytorów do konstruowania własnych ograniczeń, eliminujących te spośród kombinacji dokonywanych przez komputer, które można z góry uznać za niewłaściwe. Przykład takiego algorytmu z ograniczeniami może być wzorowany na opisie podanym przez Johna Lansdowna w broszurze „Computer Created Art" wydanej przez brytyjskie National Computer Centre.



Rys. 2.

Elementami podstawowymi będą tutaj dźwięki o wysokościach ze zbioru *c, d, e, f, g, a, h*. Bezpośrednie przejście do sąsiedniej wysokości nazwiemy „krokiem", przejście do innych wysokości „skokiem" Np. krok z *e* w górę daje *f*, *w* dół daje *d*, natomiast skok o 2 w dół daje *c*, o 3 w górę *a*. Komponując melodię startujemy od ostatniej nuty posuwając się do tyłu przy zachowaniu dwu ograniczeń:



Rys. 3.

1. Krok w danym kierunku jest rzadko poprzedzany przez skok w tym samym kierunku, częściej przez krok, zazwyczaj jednak poprzedza go krok lub skok w przeciwnym kierunku albo zmiany wysokości (ta sama wysokość może występować kilkakrotnie pod rząd w dowolnym miejscu).
2. Skok w danym kierunku nigdy nie jest poprzedzany przez skok w tym samym kierunku.

Z tych ograniczeń korzysta się w praktyce zgodnie z tabelą, przykład takiej tabeli mamy na rys. 1.

Jeśli zatem dwa ostatnie dźwięki miały wysokości *f, g* (krok w górę), a wyrzucono cyfrę 5 (lub wylosował ją komputer), to trzeci dźwięk od końca będzie miał wysokość *f* (pierwszy wiersz, piąta kolumna tabeli: brak zmiany).

Przy następnej cyfrze 1 czwarty dźwięk od końca zostanie określony na wysokości *g*. Jeśli maszyna wygeneruje ciąg liczb losowych: 5, 1, 3, 4, 6, 1, 1, 5, 4, 3... to otrzymany następującą sekwencję dźwięków w utworze: *egccdefagffg*.

kr.g.	
5 b.z.	*fg*
1 kr.d.	*ffg*
3 kr.d.	*gffg*
4 sk.g.2.	*agffg*
6 kr.g.	*afagffg*
1 kr.g.	*defagffg*
1 kr.g.	*cdefagffg*
5 b.z.	*ccdefagffg*
4 sk.d.4	*gccdefagffg*
3 sk.g.2.	*egccdefagffg*

Klasyczny zapis tego wyniku przedstawia rys. 2.

W dalszej kolejności ustalamy długość każdej nuty przypisując im cyfry innego ciągu liczb losowych, uznając np. cyfrę 1 za ósemki, cyfrę 2 za ćwierćnuty, 4 za półnuty. Jeśli więc ciąg wygenerowanych przez komputer będzie miał postać: 4, 3, 6, 6, 1, 3, 3, 4, 5, 2, 4, 1... to zapis przedstawia się tak jak na rys. 3.

Przykład powyższy, mimo, że uproszczony i zredukowany do granic niemal elementarnych, wskazuje chyba wyraźnie, jak złożone i teoretycznie trudne w realizacji jest tworzenie komputerowych kompozycji. Wbrew cytowanej w poprzednim numerze sentencji umieszczonej w Oxford Companion to Music nie da się tego z pewnością określić jako „metody komponowania bez wysiłku", nie powinno być zatem uważane jedynie za „osobliwość muzycznego życia".

Marek Hołyński

Stan obecny	Wyniki rzutu					
	1	2	3	4	5	6
krok w górę	kr. g.	kr. d.	kr. d.	sk. d. 2	b.z.	b.z.
krok w dół	kr.d.	kr.g.	kr.g.	sk.g.2.	b.z	b z
skok w górę	kr d	kr d	sk d 2	sk d 3	sk d 2	kr g
skok w dół	kr g.	kr.g.	sk.g.2.	sk.g.3	sk.g.2	kr.d.
brak zmiany: po kroku w górę	kr.d.	sk.d.2.	sk.d.3	sk.d.4	kr.d.	kr.d.
po kroku w dół	kr.g.	sk.g.2	sk.g.3	sk.g.4	kr.g.	kr.g.

Rys. 1.

J. Cage

Ogólne problemy organizowania bazy danych w systemie DMS-2

Ogólnie o bazie danych

Dotychczasowe systemy EPD były tak definiowane, że wymagały danych ściśle dostosowanych do poszczególnych zastosowań. W efekcie uzyskiwano szybkie wdrożenie pojedyńczych systemów lecz bez wzajemnej korelacji danych i bez możliwości ich wspólnego wykorzystania.
Wspólna baza danych (BD) pozwala na wykorzystanie tych samych danych do różnych zastosowań (systemów).
Najważniejsze zalety bazy danych to:
— eliminacja dublowania danych
— zabezpieczenie procesu zbierania danych przed dublowaniem
— wyeliminowanie dublowania procedur aktualizacji
— poszerzenie zasięgu systemów
— optymalna gospodarka danymi
Elementy, które sprawiają najwięcej kłopotów to:
— potrzeba specjalnego oprogramowania
— wysokie koszty początkowego tworzenia
— konieczność rygorystycznej kontroli danych wprowadzanych do BD
— potrzeba normalizacji BD
— zmiana zakresu prac analityków i projektantów systemów EPD.
Zależnie od struktury i sposobu dostępu, można wyodrębnić cztery podstawowe rodzaje bazy danych:
— zorientowaną na zbiory np.: LAMIS
— wielokluczową
— tablicową ROBOT (*tabular files*)
— kompleksową z wzajemnymi powiązaniami np. PLUTO IMS (*complex files with relationship pointers*).

Wybór koncepcji organizacji BD

Tworzenie BD nawet przy wykorzystaniu standarodowego pakietu programów, nie rozwiązuje automatycznie wszystkich problemów związanych ze zbieraniem danych, utrzymywaniem bazy i jej wdrożeniem. Za projektem BD kryje się nie tylko technika komputerowa, ale też cały szereg problemów organizacyjnych. Budowa BD jest zamierzeniem pracochłonnym i należy rozważyć możliwość jej tworzenia i wykorzystywania w kilku krokach.
Zrealizowanie nawet częściowej BD może dać również efekty, choćby tylko przez fakt centralnego zbierania i kontroli danych.
Przy wyborze koncepcji organizacji BD należy rozważyć czy korzysta się z gotowego pakietu programów, czy też będzie się projektować i programować samodzielnie.

Należy również określić cel projektowanej BD, jej podstawowe funkcje i na tej podstawie odpowiedzieć na następujące pytania:
- Jakie będzie przeznaczenie i system zarządzania projektowaną BD?
- Czy BD będzie charakteru ogólnego, jak BDMS czy bazą specjalistyczną jak DMS-2 i MIMMS?
- Czy do zarządzania BD będzie wykorzystywany system samowystarczalny (*self contained*), przeznaczony do sformalizowanej obsługi BD, jak DMS-2, czy też będzie oparty o jeden z języków wyższego rzędu, jak np. DMS-1100?

W przedsiębiorstwach przemysłu maszynowego stosuje się specjalistyczną BD do zarządzania i sterowania produkcją.
Natomiast do zarządzania BD pozornie atrakcyjne jest rozwiązanie oparte na wykorzystywaniu języków wyższego rzędu ze względu na ich szeroką znajomość.

Do podstawowych wad takiego rozwiązania można zaliczyć:
— ograniczenie możliwości manipulacji danymi wynikające z ograniczeń każdego języka uniwersalnego,
— mniejsza wydajność w porównaniu do wyspecjalizowanych systemów
— konieczność zatrudnienia programistów.

Systemy samowystarczalne (samookreślające) zostały zaprojektowane do specjalistycznych zastosowań. Mają one narzuconą własną strukturę fizyczną i własny specjalistyczny język.

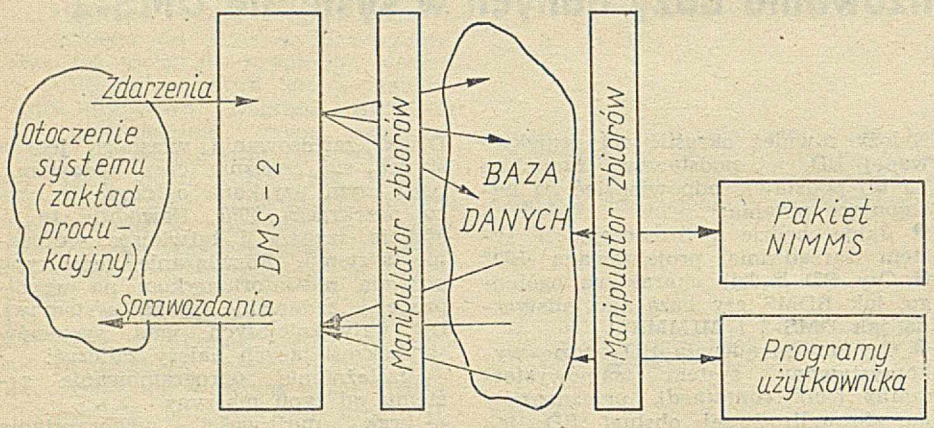
Dzięki zastosowaniu własnego języka skraca się znacznie czas kodowania, przy czym uzyskana oszczędność czasu przekracza 50%. Powoduje to w efekcie szybsze i łatwiejsze wdrożenie systemu. Rozwiązanie tego typu pozwala nie-informatykom na współpracę i zadawanie pytań systemowi.
Do najważniejszych wad systemów samookreślających należy zaliczyć:
— uzależnienie oprogramowania systemu od typu maszyny
— brak możliwości wykorzystania BD przez inne systemy i języki
— brak możliwości rozwiązywania problemów innego typu niż zdeterminowanych przez system.
— konieczność nauki nowego języka przez użytkownika.
W Zakładach Radiowych im. M. Kasprzaka (ZRK) jak wiadomo podjęto decyzję utworzenia bazy danych do zarządzania i sterowania produkc
ZRK jest Zakładem typowym dla Zjednoczenia UNITRA i dzięki temu doświadczenia zdobyte w ZRK be
mogły stanowić podstawę dla tworzenia BD w całym Zjednoczeniu.
Rozważając cel i rolę jaką musi pełnić BD w ZRK zdecydowano na skorzystanie z gotowego pakietu programów do tworzenia i utrzymywania bazy danych — DMS-2.
Do zarządzenia i sterowania produkcją zostanie wykorzystany pakiet NIMMS.

Pakiet DMS-2

Brytyjska firma komputerowa ICL proponuje organizowanie BD w o-

Kartoteki utworzone w nowej bazie danych	Wykorzystywane w systemach	Odpowiedniki kartotek w starej BD
Główna w systemie PLUTO Strukturalna w systemie PLUTO	DMS2/NIMMS	podobne funkcje TECHNO I RYSKON
Zapotrzebowań wejściowych	NIMMS	brak odpowiednika
Stanowisk roboczych (load centres file)	NIMMS	brak odpowiednika
Zleceń i zapasów (stock file)	NIMMS	podobne funkcje TECHNO, KARZAP
Operacji (part entity)	DMS2/NIMMS	KARMAT
Kosztów	DMS2/NIMMS	brak odpowiednika
Dostawców	DMS2/NIMMS	brak odpowiednika
Klientów	DMS2/NIMMS	brak odpowiednika
Płac	DMS2/NIMMS	brak odpowednika
Analiza zapasów	Systemy własne	KARMAT
Indeksów materiałowych	Systemy własne	INDMAT
Maszyn i urządzeń	Systemy własne	KASROT

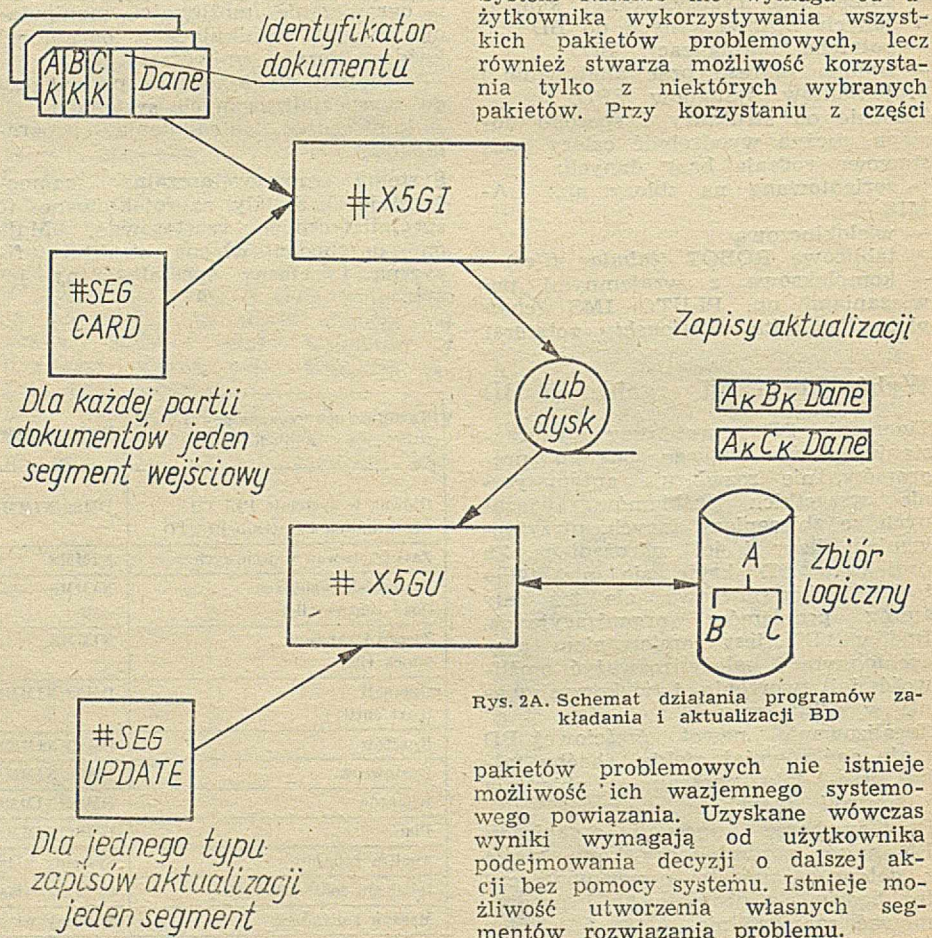
parciu o pakiet programowy Data Management System Stage 2 (DMS-2) por. rys. 1. DMS 2 składa się z programów, które umożliwiają zakładanie i aktualizację bazy danych oraz zadawanie pytań i uzyskiwanie odpowiedzi w postaci sprawozdań. Zbudowany jest z następujących podstawowych programów wejścia X5GI
aktualizacji X5GU
zapytań X5GE
sprawozdań X5GR



Rys. 1. Model działania bazy danych w systemie DMS-2, powiązań z pakietem NIMMS i z programami własnymi

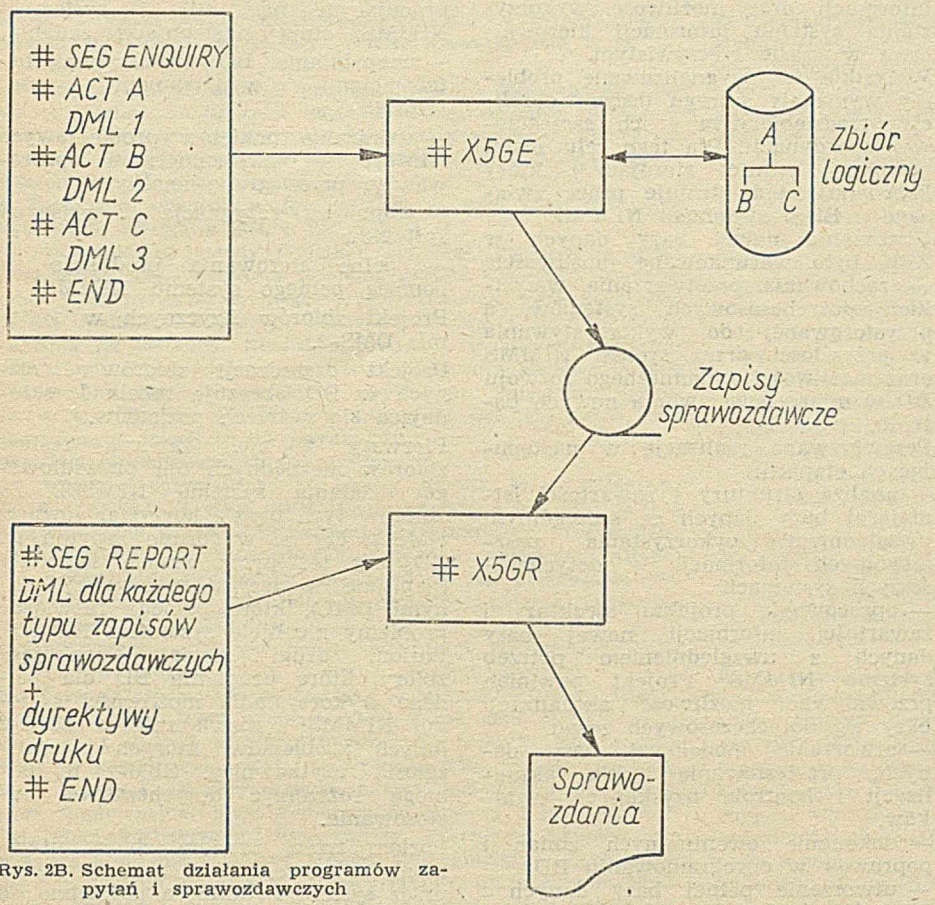
Strukturę wzajemnych powiązań i współzależności przedstawiono schematycznie na rys. 2 (AiB).
System ten jest bardzo elastyczny, gdyż stwarza możliwości stosowania również własnych rozwiązań przez tworzenie dodatkowych segmentów i pisanie programów w języku manipulacji danymi DML (*Data Management Language*). Istnieje dowolność formatu dokumentów wejścia i wyjścia.
Dokumenty drukowane analizy i sprawozdania uzyskiwane w systemie projektowane są w całości przez użytkownika i pod całkowitą jego kontrolą (X5GR). Dla emisji tych dokumentów nie stworzono żadnego gotowego programu.
Dużą pomocą w wyszukiwaniu informacji w BD jest moduł zwany manipulatorem zbiorów (*File Handler*), który zewala użytkownikowi na napisanie własnych programów bez konieczności szczegółowego zapoznania się z budową zbiorów danych. Stwarza to możliwość wykorzystywania bazy danych przez różnych użytkowników np. z różnych wydziałów bez znajomości całego problemu.
DMS-2 na podobnej zasadzie dopuszcza zmianę zawartości i struktury BD bez większego wpływu na funkcjonujące już segmenty i programy.
Dzięki temu możemy wprowadzać nowe dane i rozszerzać zakres przetwarzania, co jest nieuniknione w funkcjonującym systemie.
Dla potrzeb DMS-2 stworzono nową organizację zbiorów tzw. zbiory uogólnione. Ich cecha charakterystyczna jest występowanie w każdym zapisie (rekordzie) klucza nadrzędnego oraz adresu kontynuacji.

Obok zbiorów uogólnionych, DMS-2 korzysta ze zbiorów zorganizowanych w systemie PLUTO (*Partis-Listing Used-on Technique*). Pakiet programowy PLUTO może być wykorzystywany niezależnie od DMS-2 i jest przeznaczony do tworzenia kompleksowej bazy danych z wzajemnymi powiązaniami. Używając go można dokonywać rozwinięć (ustalać składniki (lub zwinięć) ustalić elementy nadrzędne składników).
Zbiory PLUTO można tworzyć przez pisanie własnych programów w jednym z powszechnie stosowanych języków programowania (PLAN, COBOL, FORTRAN) lub przez użycie odpowiednich istniejących już programów z pakietu DMS-2.
Ten drugi sposób jest ze względów ekonomicznych szczególnie zalecany przez specjalistów firmy ICL.
Utworzenie BD w systemie DMS-2 stanowi podstawę do wykorzystania systemu NIMMS (*Nineteen hundred Integrated Modular Management System*) jako uniwersalnego narzędzia do kompleksowego sterowania przedsiębiorstwem przemysłowym, System ten przewyższa mało doskonały pakiet PROMPT swoją elastycznością i wydzieleniem pakietów problemowych, które mogą być niezależnie wykorzystane. W systemie NIMMS istnieją następujące podstawowe pakiety problemowe:
1) Planowanie potrzeb: NIMMS *Reqeirements Planing*
2) Planowanie operatywne: NIMMS POWER (*Pert Oriented Work Scheduling Evaluation Routine*),
— harmonogramowanie zadań
— dokumentacja warsztatowa
— kontrola postępu prac
3) Kontrola zapasów: NIMMS-*Stock Control*
4) Kontrola zakupów: NIMMS-*Purchase Control*
5) Tworzenie zbioru zleceń: NIMMS-*Due Date Aggregation*
6) Kontrola kosztów: NIMMS-*Factory Cost Control*
Dzięki zastosowaniu dysków i wydzieleniu BD, NIMMS zyskał na operatywności, uzyskał łatwość aktualizacji i znacznie skórcił czas przetwarzania.
System NIMMS nie wymaga od użytkownika wykorzystywania wszystkich pakietów problemowych, lecz również stwarza możliwość korzystania tylko z niektórych wybranych pakietów. Przy korzystaniu z części pakietów problemowych nie istnieje możliwość ich wzajemnego systemowego powiązania. Uzyskane wówczas wyniki wymagają od użytkownika podejmowania decyzji o dalszej akcji bez pomocy systemu. Istnieje możliwość utworzenia własnych segmentów rozwiązania problemu.



Rys. 2A. Schemat działania programów zakładania i aktualizacji BD

Decyzja wyboru i wykorzystania pakietu NIMMS powinna być poprzedzona analizą stanu organizacyjnego i potrzeb przedsiębiorstwa.
Na podstawie tej analizy i w oparciu o możliwości systemu NIMMS użytkownik decyduje, które problemy rozwiązane zostaną w oparciu o pakiet DMS-2, a które o NIMMS.
Systemy DMS-2 i NIMMS można stosować na komputerach ICL serii — 1900 oraz ODRA serii — 1300, wyposażonych w pamięci dyskowe oraz dysponujących obszarem dostępnej pamięci operacyjnej o pojemności co najmniej 32 K słów.
W ZRK podjęto przed rokiem prace nad utworzeniem bazy danych w oparciu o system DMS-2 i stworzenia warunków do przyszłościowego wdrożenia NIMMS.
Prace prowadzone są przy udziale specjalistów z biura PROMASZ i brytyjskiej firmy ICL-DATASKIL.

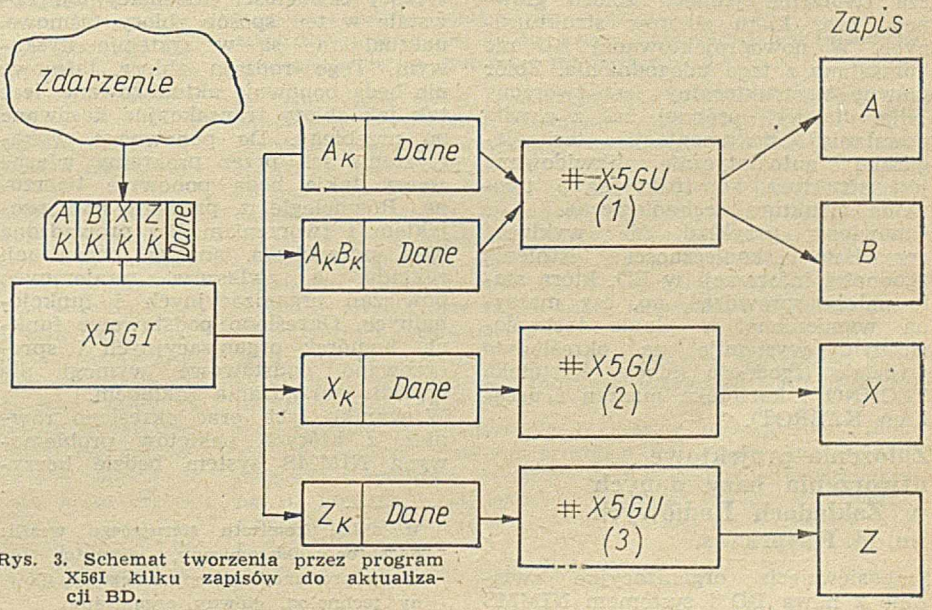


Rys. 2B. Schemat działania programów zapytań i sprawozdawczych

Aktualizacja bazy danych

W ZRK uruchomiono w latach poprzednich kilka własnych systemów zarządzania i sterowania produkcją w oparciu o istniejącą bazę danych zorganizowaną na taśmach magnetycznych.
Do najważniejszych z nich można zaliczyć:
— system planowania zapotrzebowań (między innymi obliczenie ilościowych norm zużycia zamawianych u dostawców materiałów i części)
— system planowania operatywnego (działający w systemie miesięcznym)
— system obrotu materiałowego
— system obliczania kosztów
Systemy powyższe wykorzystują sześć podstawowych kartotek BD:
1) rysunków konstrukcyjnych (RYSKON)
2) planów technologicznych (TECHNO)
3) indeksów materiałowych (INDMAT)
4) stanów magazynowych (KARMAT)
5) maszyn i urządzeń (KASROT)
8) stanów zapasów produkcji własnej (KARZAP)

Z informacji zawartych w BD do aktualnego przetwarzania tworzone są zbiory transakcyjne. W praktyce nie jest stosowana aktualizacja zbiorów transakcyjnych. Tworzenie dla każdego nowego przebiegu aktualnej taśmy transakcyjnej nie zawsze byłoby ekonomiczne, ponieważ czas jej tworzenia znacznie przekracza czas przewidziany na realizację całego problemu.
W systemie DMS-2 istnieje możliwość aktualizacji zbioru przed właściwym procesem przetwarzania. Organizacja BD w systemie dyskowym gwarantuje, że przebieg aktualizujący (X5GU) nie trwa długo. Dzięki temu do przetwarzania problemowego przekazywane są tylko informacje najbardziej aktualne.
Aktualizacja zbiorów podstawowych BD w systemie taśmowych jest również czasochłonna z następujących powodów:
— występowanie identycznej informacji w kilku kartotekach,
— wielokrotne występowanie tej samej informacji w jednej kartotece.
Aktualizacja wymaga każdorazowego przekopiowania całej kartoteki.
Występowanie tej samej informacji w kilku zbiorach przy organizacji taśmowej jest konieczne. Wymaga to jednak kilku pracochłonnych przebiegów aktualizujących BD. Przykładowo jednostka miary występuje w zbiorach INDMAT, RYSKON, TECHNO KARMAT i dla jej zaktualizowania należy wykonać cztery przebiegi.
W systemie DMS 2 zbiory fizyczne określone są tylko jeden raz i w zasadzie nie zachodzi konieczność kilkakrotnego powtarzania tej samej informacji. Jeśli nawet zaistnieje taki przypadek, to istnieje możliwość utworzenia z jednego dokumentu kilku rekordów aktualizacji. (rys. 3).
Wielokrotność występowania tej samej informacji w zbiorach taśmowych eliminuje pakiet programowy PLUTO dzięki tworzeniu dwóch zbiorów — zbioru głównego oraz zbioru strukturalnego.
Pierwszy z nich zawiera opisowe (występujące tylko jeden raz), natomiast



Rys. 3. Schemat tworzenia przez program X56I kilku zapisów do aktualizacji BD.

drugi strukturę ich wzajemnych powiązań. Przy aktualizacji wystarczy dokonanie zmiany tylko w jednym rekordzie (bez potrzeby kopiowania) i nie zachodzi potrzeba dalszego poszukiwania zbioru.

Zgodność kartotek w bazie danych

W bazie danych zorientowanej na zbiory zachodzi konieczność zgodności informacji w zbiorach. Brak tej zgodności może przyczynić się do poważnych błędów.

Na podstawie kartoteki rysunków konstrukcyjnych RYSKON i kartoteki planów technologicznych TECHNO omówiony zostanie brak zgodności struktur pomiędzy tymi kartotekami. Kompletna kartoteka rysunków konstrukcyjnych RYSKON jest uzupełniana informacjami o nowym wyrobie w czasie, gdy przechodzi on próby modelowe. Wykonanie serii próbnej odbywa się więc już w oparciu o kartotekę RYSKON zawierającą również elementarne informacje technologiczne. Jednocześnie opracowuje się technologię nowego wyrobu uzupełniając kartotekę TECHNO informacjami z planów technologicznych nowego wyrobu.

Struktura nowego wyrobu w kartotece RYSKON jest odmienna od struktury wyrobu w kartotece TECHNO. Dzieje się tak między innymi dlatego, że w procesie produkcyjnym do wyrobu jako składniki wchodzą materiały technologiczne lub wprowadzane są zmiany struktury wynikające z potrzeb technologii.

Jeżeli części do wyrobu są wykonywane jednocześnie w dwu różnych wydziałach, wtedy często może wystąpić również odmienna struktura technologiczna wyrobu.

Problemy te rozwiązuje skutecznie zasada tworzenia kartoteki rysunków konstrukcyjnych o strukturze technologicznej.

Projektując nową bazę danych w systemie DMS 2 należało rozwiązać również współzależność zbiorów RYSKON i TECHNO.

Zastosowanie pakietu programowego PLUTO do tworzenia zbioru głównego i strukturalnego wyrobów, dopuszcza tworzenie jednego zbioru głównego oraz kilku zbiorów strukturalnych. W nowoprojektowanej BD nie korzystano z tego udogodnienia. Zbiór główny i strukturalny jest tworzony tylko dla serii próbnej, zaś z chwilą powstania dokumentacji technologicznej automatycznie likwidowana jest struktura konstrukcyjna i tworzona struktura technologiczna.

Omówiony przykład nie wyklucza oczywiście konieczności istnienia zgodności informacji w BD, którą stale należy sprawdzać, np. czy maszyna wymieniona w planie technologicznym występuje na określonym wydziale (zgodność między kartoteką TECHNO i kartoteką maszyn i urządzeń KASROT).

Założenia projektowe utworzenia bazy danych w Zakładach Radiowych im. M. Kasprzaka.

Przedsięwzięcie organizacyjne związane z nową BD i systemem NIMMS obejmuje swoim zasięgiem cały zakład. Zmiany ilościowe i jakościowe informacji będą miały istotny wpływ na funkcjonowanie przedsiębiorstwa. Zastosowanie urządzeń monitorowych i terminali (ICL-7181 i ICL-7503) spowoduje szybszy przepływ informacji oraz możliwość wykorzystania systemu informacji kierownictwa w czasie rzeczywistym.

Wszystkie te zasygnalizowane problemy wymagały szeregu decyzji dyrekcji przedsiębiorstwa i ich szczegółowej koordynacji. Dla tego celu został powołany komitet sterujący [2]), który koordynuje i kontroluje prace związane z BD i systemem NIMMS.

Utworzenie nowej bazy danych w ZRK było warunkowane możliwością
— zachowania przetwarzania wszystkich dotychczasowych systemów, a przystosowanej do wykorzystywania w przyszłości przez system NIMMS oraz możliwości dynamicznego rozwoju BD w miarę powstawania nowych potrzeb.

Przewidywano realizację w następujących etapach:
— analiza struktury i zawartości istniejącej bazy danych ze szczególnym uwzględniem wykorzystania poszczególnych informacji w dotychczasowych systemach
— opracowanie projektu struktury i zawartości informacji nowej bazy danych z uwzględnieniem potrzeb systemu NIMMS. Projekt powinien przewidywać możliwość aktualizacji bazy wg dotychczasowych zasad
— utworzenie modelowej bazy danych, przetestowanie cyklu aktualizacji i kontrola uzyskanych wyników,
— dokonanie ewentualnych zmian i poprawek w oprogramowaniu BD
— utworzenie pełnej bazy danych i jej robocza eksploatacja.

Nowa dyskowa baza danych utworzona zostanie w oparciu o istniejącą już bazę na taśmach magnetycznych przez konwersję zbiorów taśmowych na zbiory dyskowe i będzie jedynym aktualnym źródłem informacji.

Zachodzi więc konieczność odtworzenia na taśmach magnetycznych zbioru o dotychczasowej strukturze i identycznej zawartości informacyjnej. Powstałe w ten sposób zbiory taśmowe, uaktualniane są w systemie dyskowym. Tego rodzaju zbiory taśmowe nie będą ponownie aktualizowane, lecz tak jak zbiory transakcyjne kasowane po przebiegu. Do ponownego wykorzystania ich przez programy własne zbiory takie będą ponownie tworzone. Równolegle z pracami nad projektem i tworzeniem BD prowadzona była szczegółowa analiza organizacji zakładu, a zwłaszcza wzajemnych powiązań organizacyjnych i funkcjonalnych. Określono podstawowe funkcje komórek organizacyjnych i sprecyzowano podstawowe wymogi dla systemu zarządzania zakładem.

W wyniku tych prac określono również, z których pakietów problemowych NIMMS system będzie korzystać, jakie problemy rozwiązane zostaną w systemie DMS-2 i jakie istnieją potrzeby rozwoju systemu w przyszłości. Wnioski z wykonanych prac zostały zawarte w opracowaniu pt. „Raport studium wstępnego". Z raportu wynika, że przed zespołem pracującym nad BD i systemem NIMMS stoją następujące zadania:
— uzupełnianie BD brakującymi informacjami wymaganymi przez NIMMS
— wdrażanie pakietów problemowych NIMMS wg harmonogramu opracowanego przez grupę analityków
— stopniowa rezygnacja z własnych systemów
— pełne sterowanie produkcją za pomocą pełnego systemu NIMMS

Projekt zbiorów fizycznych w systemie DMS-2.

Projekt organizacji zbiorów fizycznych w BD obrazuje tabela I, składająca się z trzech poziomów.

Pierwszy z nich zawiera zestaw zbiorów niezbędnych dla prawidłowego działania systemu NIMMS. Ze zbiorów tych będzie korzystał również DMS-2, czy to w formie zapytań do BD, czy też przez własne segmenty (napisane w języku manipulacji danymi DML), które pomogą rozwiązać problemy nieobjęte systemem NIMMS. Poziom drugi zawiera brakujące zbiory, które uzupełnia BD dla pełnego wykorzystania możliwości systemu NIMMS i rozwiązania w DMS-2 innych problemów, których potrzeby zgłosili użytkownicy. Zbiory te pomogą rozszerzyć dotychczasowe zastosowania.

Poziom trzeci zawiera zbiory utworzone w nowej BD i nie wymagane przez system NIMMS, a potrzebne do konwersji zwrotnej przy tworzeniu zbiorów taśmowych do wykorzystania w dotychczasowych systemach.

Jeżeli w trakcie eksploatacji wyniknie brak zapotrzebowania na te zbiory, zostaną one zlikwidowane z chwilą rezygnacji z własnych systemów.

* * *

Ta informacja o nowoprojektowanej BD nie jest w stanie oddać złożoności problemów, z którymi zespół aktualnie boryka się i które jeszcze pozostały do rozwiązania. Sygnalizując niebagatelny problem projektowania bazy danych warto pamiętać, że nie jest to sztuka dla sztuki, a działanie, które może dać poważne efekty ekonomiczne.

W Zakładach Radiowych im. M. Kasprzaka określono następujące cztery główne cele, które powinien zrealizować nowy system i zamierzone efekty:
— obniżenie poziomu zapasów materiałów produkcyjnych o 30%
— podniesienie stopnia wykorzystania maszyn i urządzeń o 12%
— zmniejszenie poziomu godzin nadliczbowych o 50%
— zmniejszenie pracochłonności prac biurowych o 25%.

Andrzej Kuroń
Zakłady Radiowe im. M. Kasprzaka
Warszawa

[2]) W skład komitetu sterującego weszli: dyrektor ekonomiczny, dyrektor do spraw produkcji, główny księgowy, główny technolog, główny organizator.

Bibliografia wydawnictw polskich z dziedziny informatyki

● **Optymalizacja pamięci wielopoziomowej. Analiza rozwiązań programowych** — DRYZEK H. Wyd. PWN, Warszawa 1974, s. 67, cena zł 20,—
Organizacja wielopoziomowej pamięci. Kryteria optymalizacji. Dobór optymalnej konfiguracji wielopoziomowej pamięci. Problemy dyslokacji bloków i analiza zachowania się programów. Algorytmy wymiany. Optymalny rozmiar strony. Słowniczek. Materiały przeznaczone są dla programistów.

● **Symulacja cyfrowa w języku CSL** — KONDRATOWICZ L. Wyd. PWN, Warszawa 1974, s. 156, cena zł 36,—
Wprowadzenie do metody symulacji cyfrowej. Podstawowe własności języka CSL. Struktura formalna modelu. Struktura funkcjonalna programu symulacyjnego. Proste instrukcje warunkowe. Cykliczna realizacja instrukcji. Nieuwarunkowane i uwarunkowane działania na zbiorach. Nieuwarunkowana i uwarunkowana kontrola stanu zbioru. System wejścia—wyjścia. Przykładowe modele symulacyjne. Systemy operacyjne języka CSL. Załączniki: dyskowa wersja CSL dla komputerów LCL serii 1900, skorowidz słów zastrzeżonych trzech wersji języka CSL.
Materiały przeznaczone są dla programistów.

● **Algorytmy optymalizacji w języku ALGOL 60** — KUCHARCZYK J. SYSŁO M.-Wyd. PWN, Warszawa 1974, s. 263, cena zł 34,—
Programowanie liniowe. Programowanie dyskretne. Programowanie dynamiczne. Teoria grafów. Analiza sieci. Algorytmy. Alfabetyczny skorowidz algorytmów. Celem książki jest przekazanie czytelnikowi zbioru pewnych algorytmów, obejmującego podstawowe metody optymalizacyjne.
Książka przeznaczona jest przede wszystkim dla użytkowników komputerów, korzystających z metod optymalizacyjnych.

● **Półprzewodnikowe układy logiczne** — MISIUREWICZ P., GRZYBEK M. — Wyd. WNT, Warszawa 1974, s. 299, cena zł 35,—
Elementy i układy cyfrowe: podstawowe techniki realizacyjne, elementy kombinacyjne TTL, przerzutniki TTL, układy pośredniczące, układy i elementy czasowe, układy pomocnicze.
Bloki funkcjonalne: bloki komutacyjne, rejestry i pamięci, liczniki, bloki arytmetyczne, komparatory. Przykłady zespołów cyfrowych: realizacje zastępcze bloków funkcjonalnych, układy pomocnicze, układy czasowe, generatory ciągów, przetwarzanie dwójkowo-dziesiętne i dziesiętno-dwójkowe, operacje arytmetyczne itd.
Materiały przeznaczone są dla projektantów cyfrowych układów automatyki i komputerów.

● **Nowoczesne maszyny i systemy cyfrowe** — Tłum. wyd. ang z roku 1971 — EADIE D. — Wyd. WNT, Warszawa 1974, s. 439, cena zł 83,—
Wprowadzenie do systemów cyfrowych. Arytmetyka i sterowanie. Pamięć i zespół wejścia-wyjścia. Organizacja maszyn cyfrowych. Programowanie i oprogramowanie. Język FORTRAN. Przetwarzanie danych. Systemy przetwarzania na bieżąco. Systemy z podziałem czasu. Transmisja danych cyfrowych. Projektowanie systemu i wybór maszyny cyfrowej. Tendencje rozwojowe. Słownik.
Książka przeznaczona jest dla czytelników, którzy ukończyli elementarny kurs podstaw projektowania maszyn cyfrowych lub mają pewne ogólne wiadomości z tej dziedziny i czują potrzebę dalszego studiowania tego przedmiotu.
Celem jest podanie wszystkiego, co inżynierowie-projektanci systemów powinni uwzględniać podczas opracowywania założeń systemu cyfrowego.

● **Metodyka projektowania systemów informatycznych** (Wzory formularzy standardów dokumentacji projektowej, programowej i eksploatacyjnej oraz ich opis) — ZYGLER H. — Wyd. Ośrodka Badawczo-Rozwojowego Informatyki, Warszawa 1974, zł 72,—
Materiały szkoleniowe. Kurs projektowania systemów EPD.

● **Wielodostępny system liczący poznańskiego środowiska naukowego. Organizacja, technika, użytkowanie** — KIERZKOWSKI Z. — red. Wyd. Politechniki Poznańskiej, Poznań 1974, s. 71.
Przesłanki i założenia struktury systemu cyfrowego. Struktura systemu liczącego. Technologia przetwarzania danych w systemie. Szkolenie użytkowników. Słownik. Sprzętowa realizacja systemu WSL w roku 1975. Opracowanie stanowi wstępną informację o przedsięwzięciu realizacji WSL, podsumowującą prace organizacyjne i techniczne.
Materiały przeznaczone są dla pracowników wyższych uczelni i instytutów naukowych wprowadzających nowoczesne metody, oparte o zastosowanie informatyki.

● **Kierunki zastosowań informatyki** — GACKOWSKI Z. Wyd. Instytutu Doskonalenia Kadr Kierowniczych Administracji Państwowej. Warszawa 1975, s. 15. Materiały szkoleniowe.
Wstęp. Automatyzacja przetwarzania danych na potrzeby zarządzania. Automatyzacja sterowania procesami. Automatyzacja prac zawodowych. Sprzęt informatyczny (rozwój w Polsce). Transmisja danych. Przygotowanie kadr. Ogólny system informatyczny.

● **Modelowanie cyfrowe systemów ekonomicznych** — Tłum. wyd. ang. z roku 1971 — NAYLOR T. Wyd. PWN, Warszawa 1975, s. 651, cena zł 80,—
Definicja symulacji komputerowej. Komputerowe eksperymenty symulacyjne na modelach systemów ekonomicznych. Modele stosowane w zarządzaniu. Modele ekonomiczne: modele firmy, modele przemysłowe, modele makroekonomiczne itp. Ocena zasadności modelu. Układ komputerowego eksperymentu symulacyjnego. Analiza wariancji. Losowanie sekwencyjne. Reguły przerywania. Symulacja a rozwiązania analityczne na przykładzie modeli ekonometrycznych. Model komputerowy przemysłu tytoniowego. Skutki polityk alternatywnych udzielania poszczególnym stanom mocy federalnej w dziedzinie oświaty — symulacja komputerowa. Model systemu monetarnego Stanów Zjednoczonych. Generatory liczb pseudolosowych. Generatory zmiennych losowych. Języki symulacyjne.
Materiał książki jest szeroko udokumentowanym przeglądem problemów i perspektyw zastosowań metod modelowania.
Przeznaczony on jest w szczególności dla ekonomistów i działaczy gospodarczych.

● **Transmisja danych w gospodarce europejskiej** — Tłum. wyd. ang. z roku 1973. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1975, s. 58. Europejski Program Badawczy DIEBOLDA. Zeszyt 67 (E-118 S), cena zł 92,—
Wprowadzenie: tendencje w dziedzinie transmisji danych. Transmisja danych — statystyka techniczna. Zastosowanie transmisji danych. Wyniki opracowania opierają się na odpowiedziach na kwestionariusz rozesłany do przedsiębiorstw i organizacji na terenie Europy.
Materiały przeznaczone są dla użytkowników transmisji danych i projektantów systemów informatycznych.

● **Obrady XXX Konferencji EPD DIEBOLDA,** Hamburg, 12—14 marca 1974 roku (wybrane materiały). Tłum. wyd. ang. z roku 1974. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1975, s. 111. Europejski Program Bdawczy DIEBOLDA. Zeszyt 66 (ED 30). Cena zł 92,—
Społeczeństwo informacji i szpiegostwo technologiczne — FARR R.
Pomiar i ocena sprawności — przegląd z punktu widzenia kierownictwa — FREEBREY R.
Pomiar sprawności — przykład zastosowania — BAHR O.
Monitory sprzętowe i monitory programowe — zapewnienia dostawców i doświadczenia użytkowników.
Materiały przeznaczone są dla kadry kierowniczej przedsiębiorstw oraz projektantów systemów informatycznych.

● **Zastosowanie metod symulacyjnych w średnio-okresowym planowaniu w skali branży przemysłu** — RADZIKOWSKI W. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1975, s. 100. Problemy informatyki.
Zdefiniowanie podstawowych pojęć. Zakres planowania średnio-okresowego w branży przemysłu. Tryb planowania średnio-okresowego w branży przemysłu. Modele i procedura symulacyjna dla potrzeb modułowego planowania średnio-okresowego w branży przemysłu. Oprogramowanie i środki techniczne symulacji.
Materiały przeznaczone są dla kadry kierowniczej przedsiębiorstw, ekonomistów i programistów.

● **Zasady programowania w języku PLAN** — NIWIŃSKI T. — oprac. Wyd. Towarzystwa Naukowego Organizacji i Kierownictwa. Oddział w Łodzi, Łódź 1975, s. 52.
Wprowadzenie: programowanie w języku symbolicznym, interpretacja pamięci, struktura programu itd. Proste rozkazy arytmetyczne z wynikiem wx (grupa 0), z wynikiem w pamięci (grupa 1). Rozkazy z argumentem literackim (grupa 10). Rozkazy mnożenia i dzielenia (grupa 4). Funkcje logiczne — rozkazy przesunięć (grupa 11).
Materiały szkoleniowe dla programistów.

● **Urządzenia peryferyjne maszyn cyfrowych — SCHRAMM H.F.M.** Tłum. wyd. niem. z r. 1973 WNT, Warszawa 1975, ss. 142, cena zł 21,—

Fakty podstawowe. Przetwarzanie danych we własnym ośrodku obliczeniowym. Usługowe centra obliczeniowe. Metody zbierania danych. Wpływ metody zbierania danych na wydajność maszyny cyfrowej i efektywność przedsiębiorstwa. Wybór metody zbierania danych. Znaczenie kart dziurkowanych. Zalety taśmy magnetycznej. Przykłady rozwiązań systemów rejestracji danych na taśmie magnetycznej. Czytniki dokumentów. Opłacalność bezpośredniego wprowadzania danych „Inteligentne" systemy zbierania danych. Zbieranie danych w produkcji. Rejestratory sprzedaży. Zbieranie danych dla usługowego centrum obliczeniowego. Organizacja ośrodków obliczeniowych. Książka przeznaczona dla kadry kierowniczej przedsiębiorstw oraz dla projektów i analityków systemów.

● **Przykłady systemów elektronicznego przetwarzania danych.** Skrypt uczelniany dla projektantów SEPD — WŁOCZEWSKIJ. Wyd. Akademii Ekonomicznej w Krakowie, Kraków 1975, ss. 364, cena zł. 21,— (skrypt).

Cz. I Bazowy system EPD dla przedsiębiorstw przemysłu maszynowego: ogólna charakterystyka systemu i przyjętych rozwiązań organizacyjnych, symbolika i dokumentacja stosowana, środki techniczne dla realizacji systemu, struktura systemu EPD, rozmieszczenie danych, schematy analityczne procesu przetwarzania danych, warunki organizacyjne eksploatacji, opis maszynowych operacji przetwarzania danych, koszty i korzyści z eksploatacji systemu.

Cz. II Systemy firmowe EPD, system DMS-2, system NIMMS, podsystem BOMP, system sterowania produkcji PICS, system SOPS, system PLANTYP.

Cz. III Ćwiczenia z zakresu projektowania SEPD, zakładanie z aktualizacją zbiorów danych normatywnych, rozłożenie pracochłonności wyrobów w cyklu produkcyjnym, obliczenie liczby robotników bezpośrednio-produkcyjnych itd. Opracowanie przeznaczone jest dla celów dydaktycznych — kształcenia projektantów systemów EPD. Wymaga ono od czytelnika ogólnej znajomości ekonomiki i organizacji przedsiębiorstw przemysłu maszynowego.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie DOS/JS. Pomocnicze makroinstrukcje Wejścia-wyjścia. C 13204-1** Wyd. MERA-ELWRO, Wrocław 1974, ss. 46.

Zastosowanie i wymagania makroinstrukcji zbiór-zbiór. Generowanie makroinstrukcji zbiór-zbiór. Makroinstrukcje dla wejścia i dla wyjścia. Przykłady programów z wykorzystaniem makroinstrukcji typu zbiór-zbiór. Tłumaczenie z wykonywania programów. Dodatki: zestawienie specyfikacji makroinstrukcji zbiór-zbiór, kadry i opisy błędów dla makro zbiór-zbiór. Materiały przeznaczone są dla programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie DOS/JS. Komunikaty operatorskie C 14002-1.** Wyd. MERA-ELWRO, Wrocław 1974, ss. 192.

Opisano wszystkie komunikaty adresowane do operatora i część komunikatów adresowanych do programisty. Każdy komunikat opisany jest pięcioznakowym kodem. Pod tekstem Komunikat w języku polskim występuje jego tłumaczenie na język polski podano przyczyny, które powodują jego wyprowadzenie. Opisano akcje podejmowane przez system przy wyprowadzeniu komunikatu oraz czynności, które powinien wykonać operator i programista celem poprawy błędu. Podano również standardowe akcje systemu w przypadku braku monitora. Podręcznik przeznaczony jest dla operatorów i programistów.

Opracował Jerzy Klamborowski

TECHNIKA ZAGRANICZA

czasopismo ogólnotechniczne, poświęcone zagranicznym osiągnięciom technicznym, wydawane przez WCT NOT

W ciągu roku ukazuje się 12 zeszytów, wydawanych w większości we współpracy ze znanymi na świecie wydawnictwami czasopism technicznych krajów socjalistycznych i krajów kapitalistycznych.

Każdy zeszyt

TECHNIKI ZAGRANICZNEJ

zawiera m. in.:

- artykuły zagranicznych specjalistów
- informacje o osiągnięciach nauki, techniki i przemysłu za granicą
- informacje techniczno-handlowe
- inne działy, jak np. krótkie notatki, przegląd zagranicznej prasy technicznej, lektorat języków obcych dla techników itd.

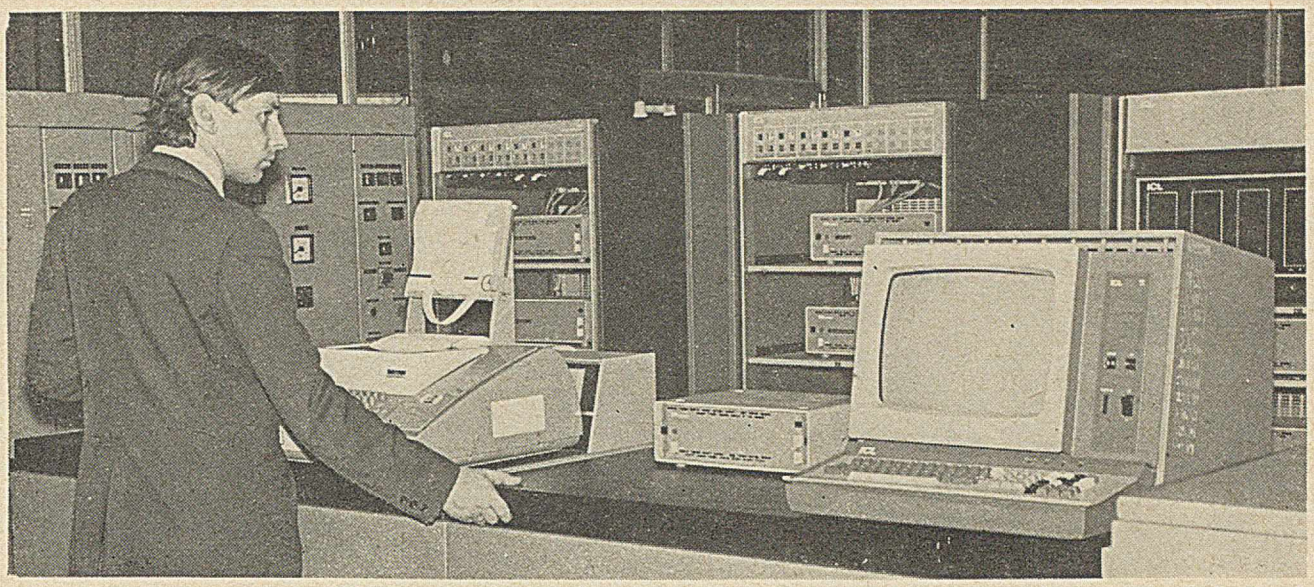
KARTA ZAPYTAŃ

znajdująca się w każdym zeszycie Techniki Zagranicznej umożliwia szybkie i bez kłopotu otrzymanie bezpłatnych informacji z zagranicy.

Cena prenumeraty rocznej 360 zł. Prenumeratę na rok 1976 prosimy wpłacać do dn. 31.10.1975 na konto:

PKO 1-9-121697
Dział prenumeraty WCT NOT
ul. Mazowiecka 12, Warszawa

W 1976 r. przewidujemy m. in. wydanie zeszytów specjalnych poświęconych technice Finlandii, Holandii, Japonii, RFN, Szwecji, USA i kilku krajów socjalistycznych.



W Zakładzie Elektronicznej Techniki Obliczeniowej we Wrocławiu pracuje (uruchomiony w IV kwartale 1974 r.) wielodostępny system POLRAX–2.

Do realizacji pracy w wielodostępnym systemie mogą być dołączone następujące terminale:

- **Dalekopisy**
  - **— zdalne 7071/7**
  - **— lokalne 7071/3**
- **Zdalne stacje ICL 7503, 7020**
- **Monitory ekranowe ICL VDU 7181/2**
- **Minikomputery RC Datapoint 2200, RC 3600**

POLRAX-2 zbudowany jest w oparciu o komputer ODRA 1305 i umożliwa korzystanie z całego oprogramowania komputerów serii ODRA 1300.

Dołączenie do systemu jest dokonywane na podstawie porozumienia lub umowy między ZETO a klientem.

Na zdjęciu urządzenia transmisji danych systemu POLRAX-2.

KOMUNIKAT

W listopadzie ub. roku odbyła się krajowa kursokonferencja naukowa zorganizowana przez Zakład Elektronicznej Techniki Obliczeniowej w Bydgoszczy i Oddział TNOiK na temat: „Organizacja systemów informatycznych w zakresie gospodarki materiałowej przedsiębiorstwa".

Powodzenie tej konferencji prowadzonej w oparciu o powielarny podsystem p.n. Ewidencja stanów i obrotów materiałowych (SEMO), opracowany w ZETO Bydgoszcz jako podstawy modułu systemu gospodarki materiałowej, skłania do kontynuacji tego cyklu konferencji.

W dniach 9 i 10 października b.r. w sali NOT Bydgoszcz ul. Rumińskiego 6 odbędzie się konferencja naukowa na temat „Projektowanie i wdrażanie powielarnych systemów informatycznych" organizowana przez TNOiK oddział w Bydgoszczy i ZETO-Bydgoszcz.

Tematyka konferencji uwzględniać będzie prezentację metod projektowania powielarnych systemów przez trzy ośrodki ZETO (Gdańsk, Wrocław, Bydgoszcz) wraz z ilustracją praktycznych rozwiązań systemów. Drugą grupą tematyczną będą problemy wdrażania systemów powielarnych. Zostaną w niej omówione pewne już zrealizowane działania w różnych organizacjach gospodarczych, które były warunkiem rozpowszechniania systemów powielarnych u większej ilości użytkowników. Metody rozpowszechniania zostaną zilustrowane przykładami z przemysłu maszynowego, chemicznego, budownictwa i obrotu lekami. (Ks)

cena zł 15.—

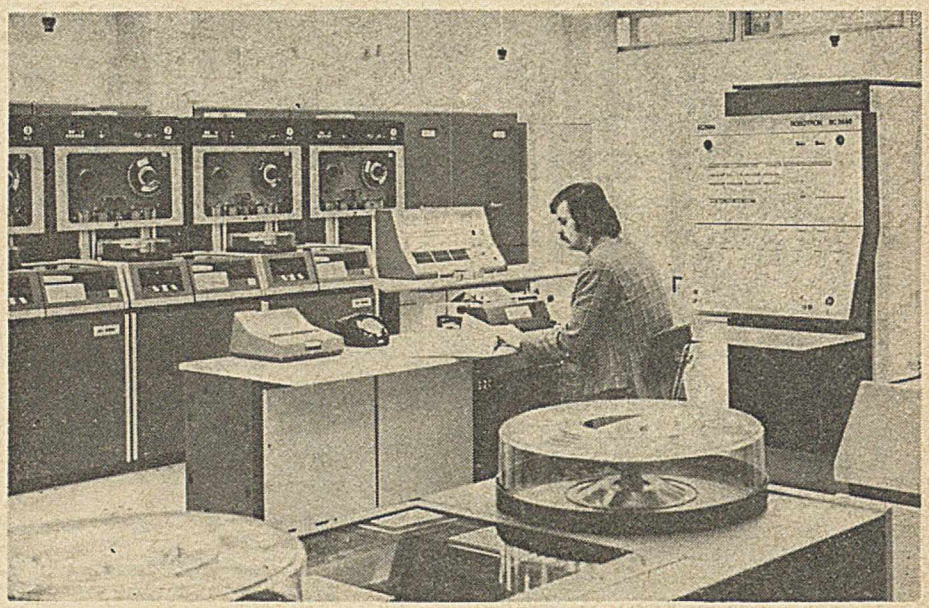
robotron

Problemy
Wymagania
Rozwiązania

Stają przed Wami podobnie, jak przed nami!

Nasza droga:

JS — Jednolity System Elektronicznych Maszyn Cyfrowych.
To symbol współpracy nie znającej granic państwowych — olbrzymi potencjał w służbie producentów i użytkowników.



Nasza oferta:

Komputer do przetwarzania danych ROBOTRON EC 1040 ze zmienną liczbą urządzeń zewnętrznych JS do rozwiązywania zadań gospodarczych i naukowo-technicznych, do zwiększania efektywności zarządzania, planowania i rozliczeń, do zastosowania jako komputer główny w systemie hierarchicznym. Oprogramowanie systemu ukierunkowane maszynowo i problemowo, szkolenie dostosowane do konkretnych wymagań użytkownika, wypróbowany serwis techniczny — to zakres świadczonych przez nas usług!

Program dostosowany do Waszych potrzeb! Korzystajcie z bardziej szczegółowych informacji!

bme

Eksporter:

**Büromaschinen-Export GmbH Berlin
DDR — 108 Berlin,
Friedrichstrasse 61
Niemiecka Republika Demokratyczna
Przedstawicielstwo BME w Polsce
Biuro Techniczno-Handlowe przy Ambasadzie NRD
Warszawa,
ul. Filtrowa 62 m. 63**

**Sprzedaż i informacje:
Biuro Generalnych Dostaw MERA-ELWRO-SERVICE
Wrocław, ul. Ostrowskiego 32**

WCT/431/K/75-A